01　組み込みアプリケーション・ソフトウェア開発者

山科祥悟

組み込みアプリケーション・ソフトウェア開発者とは

規模・目的に関係なく，専用のハードウェアに搭載されて特定の機能を実現するシステムを「組み込みシステム」，そこに搭載されるソフトウェアを「組み込みソフトウェア」と呼ぶ．こうしたソフトウェアを開発する技術者が組み込みアプリケーション・ソフトウェア開発者（以下，ソフトウェア開発者）である．ソフトウェア開発者は，組み込みソフトウェアで実現する機能の分析，設計，コーディング，テストを行う．組み込みソフトウェアには，設計条件にさまざまな制約が存在し，実現の難易度が高い．それゆえ，挑戦しがいのある分野であるとも言える注1．

● まずは，一つの知識を徹底的に理解・習得する

経験1～2年目の技術者は，C言語でいう関数の詳細設計やコーディング，関数単体テストなどを担当します．その作業に必要な知識として，以下の三つが挙げられます．

- 設計技法（例えば構造化設計．逐次・条件分岐・繰り返しの基本や処理単位を分割するSTS分割技法など）
- プログラミング言語（例えばC/C++言語．組み込みソフトウェア開発ではC言語が多用されている注2）
- テスト技法（同値分割，限界値分析，制御の流れに着目する制御パス・テストなど）

また，ソフトウェア工学（アルゴリズムの知識）や使用するOSのスケジューリング動作について知っていれば，奥の深いソフトウェアを開発することができます．ただし，知識を詰め込み過ぎては消化不良を起こし，それぞれの技法を有効に活用できません．まずは上述のどれか一つを徹底的に理解し，習得することが大事です．一つのことを理解しようとすれば，周辺の知識が必要となり，おのずとほかの事がらも理解できるようになります．

近年の組み込みシステムでは，携帯電話やカー・ナビゲーション・システムなどのように，GUI（graphical user interface）を備える製品が目立つようになってきました．しかし，これは組み込みシステムの中のほんの一部で，そのほとんどは目に見えないところで稼働しています注3．その見えないシステムやソフトウェアを開発するのですから，創造力と想像力，そして論理的思考力が求められます．

このように書くと，「自分はどうも…」と敬遠されるかもしれません．しかし，例えば，規則性を持ったソフトウェアには数列や帰納法の考えかたが適用できますし，ソフトウェア・テストは集合論を応用しています．理数系の基礎ができていれば，問題なく組み込みアプリケーション・ソフトウェア開発者になれます（**図1**）．



図1　理数系の基礎ができていればだいじょうぶ
例えば，規則性を持ったソフトウェアには数列や帰納法の考えかたが適用できる．ソフトウェア・テストは集合論を応用している．

● 世の中のために役に立ったことを実感できる

苦労する点は，さまざまな制約（メモリ量，CPU性能，リアルタイム性，高信頼性，安全性，価格，実行環境など）の中で品質を確保し，開発を完了させなければならないことです．また，ソフトウェア開発には技術的な制約のほかに，予算や工程の制約もあります．品質を高めるには手間と時間とお金がかかります．安易に妥協すると，顧客の評価が下がり，信用をなくします．しかし，高すぎる理想を追い求めると，開発者が疲弊します注4．

注1：決して，ほかの分野のソフトウェア開発がやさしいということでなく，制約条件がより厳しいという意味である．
注2：トロン協会が実施している「組込みシステムにおけるリアルタイムOSの利用動向に関するアンケート調査」（http://www.assoc.tron.org/jpn/research/index.html）などで報告されている．
注3：携帯電話，エアコン，テレビ，冷蔵庫など，見渡してみると，組み込みシステムに囲まれていることに気付くはずである．
注4：IPA（独立行政法人 情報処理推進機構）SEC（ソフトウェア・エンジニアリング・センター）の「2005年版組込みソフトウェア産業実態調査報告書」（http://sec.ipa.go.jp/download/200506es.php）などを参照．
注5：大きなシステムほど，稼働している状況を間近で見られる機会は多くない．現場の雰囲気を知り，システムを理解するためにも積極的に客先の現場に赴こう．
注6：http://www.ertl.jp/SWEST/を参照．

筆者は入社2年目に，システム・テストのため，顧客のもとに出張しました．それまでは，パソコンに向かって仕様書に記載された内容に忠実にコーディングし，テストしていただけで，自分の開発したソフトウェアがいったい何の役に立つのか，理解どころか想像すらできませんでした．ところが，そのソフトウェアは顧客のもとで宇宙機器の壮大な管制制御装置の一部として機能し，その役割を果たしていました．正常に動作しているソフトウェアをまのあたりにしたとき，自身の作ったものが世の中のシステムの中で役に立っていると実感がわき，非常に感動しました[注5]．その感動を何度も味わいたいがために，ソフトウェア開発を続けていると言っても過言ではありません．

やりがいは人それぞれだと思いますが，筆者の場合は，自分の設計したソフトウェアが世の中の役に立っていることを実感できたときに，「がんばって良かった」と感じます．

● 新人技術者への三つの質問

筆者は新人エンジニアに，以下の3点を質問します．

1）自分に投資していますか？

工業製品を開発する者としてしごとをするのですから，趣味のプログラミングとは違って体系的な専門知識が必要です．しかし，「しごとを教えてもらえる」という消極的な考えかたでは務まりません．自分に投資して，しごとの成果で回収する，という考えを持ちましょう．それが，自分自身のスキルを高め，より良い製品の開発につながります．

2）動機（モチベーション）を持っていますか？

ソフトウェア開発は知的労働です．入社1～2年目は知らないことが多く，勉強としごとの両立で時間の余裕がありません．そのときにこそ考えてほしいのは，「何のために勉強しているのか？」，「何のために働いているのか？」ということです．筆者は自分の作るソフトウェアで世の中をもっと便利に，豊かにしたいという気持ちを持って，18年前にこのしごとに就きました．短納期や技術的難易度の高いしごとでは辛い思いをしましたが，動機があったからこそしごとの山を乗り越え，達成感を持てたのだと思います．

3）ものづくりは好きですか？

組み込みソフトウェア開発に携わって，いろいろな人としごとをしましたが，ものづくりが好きな人たちほど楽しくしごとをして，良い製品を世に送り出しています．必要な知識，スキル，心がまえなど，いろいろ書きましたが，これがいちばん大事なことです．

筆者は先日，SWEST7[注6]に参加しました．日本の組み込み業界の著名な方から学生まで，幅広い技術者が参加していましたが，この人たちの共通項は，「ものづくりが好き」，「率先してみずからものづくりを楽しんでいる」ということでした．もし，「ソフトウェア開発が楽しくない」と感じ始めたら，一度参加してみることをお勧めします．

やましな・よしのり
三菱電機マイコン機器ソフトウエア（株）

2

私の推薦書

「Code Complete 第2版 ―完全なプログラミングを目指して（上・下）」



Steve McConnell 著
日経BPソフトプレス 刊，
ISBN4-89100-455-X/4-89100-456-8，B5変形判，
628/545ページ，6,405円，2005年3月

「プログラマ必携のバイブル的書籍です」と，紹介者は使い古された定型の賛辞でお茶を濁したくなります．なぜなら，1行1行から得られるものがそのくらい多いからです．

この本には，より良い製品を開発するため，高品質で高信頼性のコードを書くためのヒントが満載されています．コーディングとデバッグに主眼が置かれていますが，単なる実装論にとどまらず，上位設計からテスト手法，品質保証プロセスの重要性まで言及されています．読み進み，理解が深まるとともに，ソフトウェア開発はプログラムを書く（コーディング）だけではないということ，すなわちソフトウェア開発の奥深さを知ることができます．読み返すたびに新たな発見がありますが，「驚き」が「うなずき」に変われば，それはあなたの成長のあかしかもしれません．堅苦しい印象を受けるかもしれませんが，軽快な文章（翻訳者の努力のたまもの）が理解を手助けしてくれます．

第1版発行から第2版発行までに約10年が経過していますが，10年間のソフトウェア工学発達の成果が余すところなく投入されています．考えかたはオブジェクト指向が主体となり，言語はC++，C#，Java，Visual Basicなどが取り上げられています．このようなことからも，第2版はより近代的な印象を受けます．

すべてを理解するのはたいへんですが，全体を大ざっぱに眺めた後，コーディングの現場で目次や索引を活用して必要なときに適用すると，より実践的な知識として身に付くことでしょう（とは言うものの，1回は落ち着いて目を通さないといけないのだが…）．

02 ミドルウェア開発者

坂本直史

ミドルウェア開発者とは

ミドルウェアとは，OSやモニタ・プログラムといったソフトウェアでいちばん基本となるハードウェア資源を管理するソフトウェアと，製品仕様を主として実現する上位層のアプリケーション・ソフトウェアの間に位置する，あるまとまった機能を実現するソフトウェアである．例えば動画像処理やオーディオ処理，標準的に用いる通信プロトコル・スタックなどがある．ミドルウェア開発者は，このようなソフトウェアの設計・開発を行う．

● 技術文書を論理的に読解できる能力が必要

組み込み用途のミドルウェアの多くは，規格書になっているものを実現したり，汎用コンピュータやパソコンの世界で実用化されている既存技術を応用したものです．前者の例としてMPEG規格に準拠した動画像処理やオーディオ処理，RFCに準拠した通信プロトコル・スタックが，後者の例としてファイル・システムや画像/音声の認識，信号処理プログラムなどがあります．

規格書に書いてあるものを作るのは簡単だと思われるかもしれませんが，規格書に書いてあることを正確に理解するのはけっこう難しいものです（規格書は英語で書かれたものが多く，また日本語訳があっても正確さに欠けることがある）．そのため，ミドルウェア開発者になるには，技術文書を論理的に読解できる能力が必要だと考えます．



図2　ミドルウェア開発者の夢

ミドルウェアの開発には長い時間がかかる．規格書などを読みながら，さまざまな問題を解決していく必要がある．自分が開発したミドルウェアを搭載した製品がいろいろと街に現れるのは，大きな楽しみ．

一方，既存技術を応用するには，その技術に精通する必要があります．専門の研究者である必要はありませんが，ちょっとした専門家レベルにはなっていなければなりません．そういう意味では，新しいことを勉強したいという好奇心と，詳しく知ろうという探求心が必要だと考えます．

最後に，当然のこととしてソフトウェアの開発能力が必要です．ただし，ミドルウェア開発は「ソフトウェアが作れればそれでよい」というのとは異なります．ミドルウェアには以下のことが要求されます．

1）いろいろな製品に組み込まれて使用されるため，できるだけ多様な使いかたに対応できること
2）多くのシステムに組み込まれて使用されることが前提となるため，高い信頼性を実現すること
3）CPU性能に限りがあるため，できるだけ少ないCPU性能で処理できること
4）将来の仕様変更に対応したり，移植性を備えること

とくに1）や4）のためには，開発するソフトウェアの実際の使われかたや，どのような仕様変更が発生しそうかということを，事前に想像できる能力が必要になります．

いろいろと書きましたが，まとめるとミドルウェアの設計・開発には特別の知識やスキルが必要なわけではありません．ただし，好奇心や探求心があって，しかも論理的な思考とそれをベースとした豊かな洞察力というものが必要ではないかと思います．

● 自分の開発したものを搭載した製品が街中に現れる

組み込みミドルウェアのしごとでいちばん苦労するのは，「これが必要だ」と信じて作ってもなかなか売れないこと，そして，必要とされるときは「今すぐ欲しい」と言われることです．ミドルウェアを開発して製品化するには長い時間がかかります．長い時間をかけて製品化した以上，売れないと困ります．また，いくら必要でも，魔法のようにパッとはでき上がりません．

パソコンを対象としたソフトウェアとは異なり，組み込み用途の場合，ミドルウェアが対象とするマイコンが採用されないと（性能にいくら自信があっても），そのミドルウェアには声がかからないという歯がゆさがあります．

顧客のシステムに組み込んだときにトラブルが発生するとやはり苦労します．ミドルウェアの場合，多くの顧客に採用されることを前提に作っているので，万一不ぐあいが見つかると，すでに採用されている(もしかしたらすでに市場に出ている)案件にまで影響がでます．つまり，その影響はある特定の製品や顧客にとどまりません．

一方，しごとのやりがいは，上述の苦労する点と反対で，自分が「絶対に必要になる」と信じて開発してきたものが採用されることです．1年，2年と時間をかけて開発してきたミドルウェア，ときには周囲から認知されず，ひそかに開発を続けてきたものが採用されると，小躍りしたくなります．とくに，性能面が優れていて採用されたときは格別です．そして，自分の開発したミドルウェアを搭載した製品がいろいろと街に現れるのも大きな楽しみです．ミドルウェアは複数のメーカに採用されることが多く，その製品の種類もバラエティに富むことになります．そのため，いろいろな製品に自分の開発したものが搭載されているという状況が発生すると，楽しみは何倍にも広がります(**図2**)．

ディジタル・カメラが普及し始めたころ，自分の開発した静止画像圧縮のミドルウェアが搭載された製品が某社から発売になり，テレビで有名な女優がコマーシャルに出ていてうれしかったことをよく覚えています．そのディジタル・カメラをいまでも愛用しています．

● 画像からセキュリティまで，広く深い知識が身に付く

筆者はもう15年以上もミドルウェアのしごとに携わっていますが，そのおかげで，画像処理やオーディオ処理，ファイル・システム，通信，セキュリティなど，いろいろな技術を身に付けることができました．もちろんそれぞれの分野のエキスパートではありませんが，それぞれの技術を勉強する機会を得て，広く深い知識を身に付けることができました．ソフトウェア開発が好きで，それ以外にも好奇心や探求心があり，いろいろな技術を習得していきたいという方には非常におもしろいしごとだと思います．

このしごとを目指すには，まずソフトウェア開発の技術を身に付けてください．単にC言語でプログラムが書けるというだけではなく，コンパイラ技術やスクリプト言語など，ソフトウェア開発に役立つ関連技術も重要です．また，組み込み用途のミドルウェアには，当然，組み込みソフトウェアの技術やマイコンのアーキテクチャ，周辺回路などのハードウェアに関連する知識も必要になります．

加えて，自分が興味のある技術を，まずは自分で勉強してください．画像処理でも音声技術でも，いきなり自分の興味のあるしごとに就けることはまれですが，いつチャンスが巡ってくるかわからないからです．そして，自分の興味を膨らませていただければ，と思います．

さかもと・ただし　(株)ルネサス ソリューションズ

2

私の推薦書

"Algorithms + Data Structures = Programs"

Niklauth Wirth 著
Prentice Hall 刊，
ISBN0-13-022418-9，24cm × 16.5cm，
366ページ，1976年2月



ミドルウェア全般について解説した書籍を，残念ながら筆者は知りません．個別の技術については，それぞれ専門書があります．

このような中で，少し古いのですが推薦書としてNiklauth Wirthの著作を挙げます．ミドルウェアの設計には外部インターフェースを決めること，テストの方法を考慮して設計することなど，いろいろと重要なことがありますが，その中核は，機能をどのような手順で実現するか，つまりアルゴリズムです．いくら機能を実現できても，アルゴリズムが悪いと性能が出ません．そして，アルゴリズムと密接に関係するのがデータ構造です．アルゴリズムを効率的に実現するために，どのようにデータを保持していくかを考える必要があります．データ構造が悪くて全体の効率が下がるようだと，せっかくのアルゴリズムも生きてきません．

プログラムの基本となるアルゴリズムとデータ構造の重要さを本書の著者は"Algorithms + Data Structures = Programs"というタイトルで表現しているのだと考えます．アルゴリズムを決め，それを実現するための具体的なデータ構造を決めれば(つまりアルゴリズムという骨格にデータ構造で肉付けすると)，プログラムはおのずから表れてくるということでしょうか．

この重要さに加えて，組み込み用途のミドルウェア開発ではハードウェア資源や実行時間などの制約を考慮しながら，アルゴリズムとデータ構造の両者に必要なROM/RAM容量，処理性能，テスト容易性，拡張性などを考慮して検討していくことが重要になります．組み込みシステムでは，ハードウェア資源に限りがあります．いくらアルゴリズムが良くても，多くのRAM領域が必要になると，実現できないかもしれません．また，データの読み出しや書き込みに多くの時間がかかって，全体として効率的ではないかもしれません．こうした制約があるからこそ，みなさんの出番があるのです．

03 OS/開発ツール開発者

中島 健

OS/開発ツール開発者とは

OS/開発ツール開発者は，基本ソフトウェアであるOSや，ソフトウェア開発を支援するデバッガ，コンパイラ，ビルド・ツール，コンフィグレーション・ツール，エディタ，システム解析ツールなどを開発するエンジニアである．こうしたOSや開発環境は，ディジタル・カメラやカー・ナビゲーション・システムなどの民生機器から人工衛星に至るまで，幅広いシステムの開発を支えている．

● OSと開発ツールは密接にリンク

組み込みシステム開発でOSが使用されるのはなぜでしょうか？ 極論かもしれませんが，OSがないと開発できない機器はないはずです．それでも，とくに大規模なシステムの開発では，OSを使用するのが一般的です．

組み込みシステムは複雑化しているにもかかわらず，組み込み技術に精通した技術者は不足しています．しかも厳しい競争を勝ち抜くために，より安く，早く，そして質の高い製品を開発することが求められています．こうした状況下では，比較的経験の浅い技術者であっても，質の高い製品を開発することが求められます．OSはこうした問題に対処する方策の一つであると言えます．

OSを使用するメリットの一例として，動作検証済みのプログラムを再利用しやすくなることが挙げられます．これにより，開発期間も短縮できます．

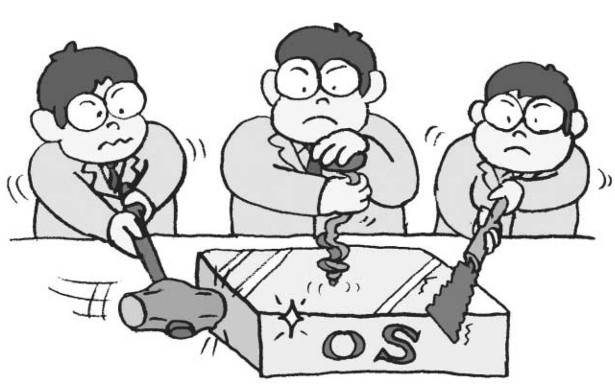


図3　OSには堅ろう性が求められる

OSには，どのような使いかたをされても仕様どおりに動く堅ろう性が求められる．アプリケーション・ソフトウェアと異なり，ユースケース（利用方法）を想定して品質や性能を取捨選択することができない．

OSの構成機能には明確な定義はありません．おそらく，カーネル，もしくはそれにファイル・システムや通信プロトコル，シェルなどを付加したものをOSと呼ぶのが一般的なのではないでしょうか．ですから，本稿においてOSと開発ツールを同じカテゴリで語ることに違和感を持たれる方がいるかもしれません．しかし，ユーザのアプリケーション開発を支援するという意味では，開発ツールはOSの構成要素です．

ところで，最近の組み込みシステム開発ではOSに関する知識が必要になることが多く，それに伴ってOS技術に精通した技術者が求められています．一方，OSの理解に時間を割く余裕のある技術者はまれで，OS技術者は不足していると言えます．開発ツールは，こうしたOSおよびOSを使用したシステムの理解を促進するものでもあります．例えば，システムの動作を外部（例えばWindowsなどのホスト・パソコン）から簡単に参照できるシステム解析ツールや，マルチタスクをサポートしたデバッガなどの機能が提供されます．

こうした機能は，ユーザが組み込みシステムを開発する際に求める利便性を考えると，必須のものと言えます．また，ソフトウェアの再利用を容易にする構成管理ツールや，最適化されたコンパイラなどと連携することで，それぞれのツールが別々に提供される場合よりも大きな利便性を実現できます．OSのカバーする分野はこれからも増えていくと思われるので，開発者はこうした連携を意識する必要があります．

● ユーザの要求を先読みできる知識や技術が必要

通常のアプリケーション（ソフトウェア）開発では，例えばディジタル・カメラであれば高速画像処理が重要であり，自動車の制御システムでは信頼性や耐故障性に重点が置かれるなど，製品に即した最適な設計が要求されます．しかし，OSの場合は事前に適用分野を限定できないため，さまざまな用途を想定したうえでの設計が求められます（こうした事情は，汎用OSか特定用途向けOSかによって程度が異なる）．ただし，限定しない結果として，OSが大きくなり過ぎたり，動作が遅くなることは，リソースの制限が厳

しい組み込みシステムでは許されません．機能をできるだけ小さくモジュール化して，スケーラブルなシステム構築を可能にするなど，拡張性のある柔軟な設計が求められます．

また，OSはシステム全体に与える影響が非常に大きいため，どのような使いかたをされても仕様どおりに動く堅ろう性が必要です．上位のアプリケーション・ソフトウェアではユースケース（利用方法）を想定して品質や性能の取捨選択ができるかもしれませんが，OSにはこれは許されません．すべてにおいて正確に高速に動作することが求められます（**図3**）．

こうした条件を満たすために開発者はどうすればよいのか，という問題に明確な答えはありませんが，技術力を磨くことはもちろんとして，さまざまな分野の知識を能動的に吸収しようという姿勢がいちばん大事なのではないでしょうか．上記の事がらは，ユーザの要求を先読みしたうえでOSを開発しなければならないということを示しています．そのためには，ユーザの問題を解決するための知識や技術が必要ですし，ユーザ以上にユーザのしごとを理解していなければならない場合もあります．

● システムの心臓部を設計・開発できる充実感

OSには複数のコンポーネントが存在するため，求められるスキルはそれぞれ異なります．また，OSがカバーする技術範囲も非常に広範に渡るため，すべてを一気に学習することは不可能です．しかし，逆に言えば，さまざまな経験を生かせる場が用意されているとも言えます．例えば，アプリケーション・ソフトウェア開発者であれば，ユーザの視点からの提案ができるでしょう．ハードウェア・アーキテクチャに詳しいのであれば，デバイス・ドライバなどの設計にその知識を生かせるでしょう．まずは自分の経験や興味と照らし合わせてターゲットとなるコンポーネントを決め，それについて勉強し，実務の中で段階的にスキルを習得していけばよいでしょう．

組み込みシステムの基幹部分であるOSには，当然高い品質が求められます．フットプリント（使用メモリ容量）やオーバヘッドなどの制限も厳しい中で，ユーザであるアプリケーション・ソフトウェア開発者の要求にこたえなければなりません．これはたいへんなことですが，システムの心臓部を設計・開発するという経験は，だれにでもできるものではありません．システムがどう動いているのかについて興味がある方にとっては貴重な体験ができるしごとと言えるでしょう．

なかしま・たけし　イーソル（株）

2

私の推薦書

「モダンオペレーティングシステム 原著第２版」

Andrew S. Tanenbaum 著
ピアソン・エデュケーション・ジャパン 刊，
ISBN4-89471-537-6，23cm×18.2cm，
986ページ，7,350円，2004年12月



この書籍は大学などで教科書として広く使用されており，OSの基礎理論をひととおり知ることができる名著です．欠点は重くて高価なことです．まずは図書館などで第1章だけでも読んでみてはいかがでしょうか．それだけでも，OSの要件をふかんすることができます．構成もよく考えられており，興味を持った章から読み進められるようになっています．また，巻末の章はUNIX/LinuxとWindowsの事例研究，およびOSの設計方法についての説明に割かれており，トレードオフや機能分割などの問題を具体的に理解することができます．これらは自分がOSを作成する場合に必要なコンセプトを考えるのに有用でしょう．

本書は基本的なことからていねいに説明してくれるため，読みやすく感じると思いますが，じっくりと読んでも得るものは多いはずです．コンピュータ業界は変化が激しい業界ですが，だからこそ基本や温故知新の精神が重要であると筆者は考えます．本書でも例示されているように，仮想マシンなどは過去に一度見捨てられたOS技術を再利用したものです（余談だが，JavaなどはOS上で動くOSと言えるのかもしれない．抽象化レイヤが異なるだけで，その思想や概念には多くの共通項が見いだせる）．ハードウェアの進歩などの環境要因によって，今日，時代遅れのアイデアが明日には花形の技術になって復活するかもしれないのです．また，本書を読めば，OSカーネルの基本的なしくみが何十年も前から驚くほど変わっていないことに気付くでしょう．そういう意味でも，しっかりとした基礎を身に付けることは非常に大事です．

最後に，筆者のもっとも印象に残っている，非常に深い示唆を含んだ（逆説的な）文章を引用します．

「膨張し，バグの多い，信頼性のないOSを排除する一つの方法は，OSをより小さくすることであるが，これより過激な方法の一つは，OS自体をも排除する方法である」．

OS開発を突き詰めていくと，（それが随分先のことだとしても）OSの存在を否定する結果になるかもしれません．ただし，そのためには，OSを十分知ったうえでOSに代わるコンセプトを発見する必要があります．あたりまえですが，技術は手段に過ぎません．本稿の冒頭でも述べたように，OSの目的を理解したうえで，技術をどう生かすのかを考えてほしいと思います．

04 ソフトウェア・テスト技術者

杉浦英樹

ソフトウェア・テスト技術者とは

ソフトウェア・テスト技術者は，出荷するソフトウェアに含まれる不ぐあいを効率良く検出することを目的に，テストの設計と実装，実施，管理を行う．まず，ソフトウェア・テストの対象となる組み込みシステムに要求される機能や性能を調べ上げる．次に，調べ上げた結果をもとに，どのようなテストを実施するかを考えて，テスト設計書を作成する．このテスト設計書に従って，具体的なテスト項目を作成し，テストする．テスト技術者は，検出した不ぐあいがどのようなものかをソフトウェア開発者に正確に伝えることで，ソフトウェアの品質作り込みの一翼を担っている．

● いつも沈着冷静，客観的に分析して報告できる性格

ソフトウェア・テストを生業(なりわい)とするときに必要となるのは，第1に，テスト対象となる組み込みシステムが「何をどのようにするシステムか」ということを分析する方法の知識や分析力です．つまり，その組み込みシステムを使う顧客の立場に立ってテスト対象をとらえ，テストを実施できることがもっとも重要です．

もともとソフトウェア・テストは，要求どおりに動くか，設計したとおりに動くか，実装したとおりに動くかということの確認行為です．テストを設計したり，テスト項目を作成するためには，何がどう動けばよいのかをしっかりと理解する必要があります．これは，テストを行うために必要なスキルですが，要求工学[(1)]のように，「だれが，何を，何のためにするのか」ということをテストというスケールで考えることになります．したがって，例えばソフトウェア開発で作成した要求分析結果をテストの入力として，「だれが，何を，何のためにするのか」の視点でどうテストするのかを考えるわけです．このように考えてテストを実践することで，ソフトウェア品質を高めるという大きな目的を達成することができます．

第2に，テスト手法に関する基礎知識が必要になります．ソフトウェアは，複雑化，肥大化の一途をたどっており，テストに対する考えかたもさまざまです．このような状況ですから，テスト手法の研究は活発で，考えかたも日々変化しているのが実情です．

テスト手法に関する基本の知識はすでに確立された方法論から学び，樹木にたとえるなら幹になる考えかたを身に付けます．対象となるソフトウェアを大地とすると，テストは根のようなものです．さまざまな領域のソフトウェアをテストした結果を共通の経験として体系化したものが，テスト手法の基本になります(**図4**)．

ソフトウェア・テスト技術者は，日々最高のテストを目指すので，基本的なテスト手法だけでは満足できなくなります．より良いテストを行うために，最新のテスト手法についてアンテナを張って，実践しなければなりません．最新のテスト手法は，中枝にあたる部分です．最新のテスト手法をどんどん使うことによって，テスト効率が上がり，テスト結果をソフトウェア品質にフィードバックできます．

テスト結果を反映したソフトウェアをさらにテストしていくことで，ソフトウェアの品質は改善され，テスト技術も発展します．このことは，ソフトウェア・テスト技術者のみに言えることではありませんが，実際に効果が期待できる理論や事例はどんどん実施してみましょう．実施すれば，くふうや改善の余地，特性や問題が明らかになり，次のアクションを楽しむことができるでしょう．

ソフトウェア・テスト技術者の性格や特性としては，「情緒に流されず，いつでも冷静に事実を客観的に分析し，報告できる」ことが求められます．商品開発も佳境に入ると，

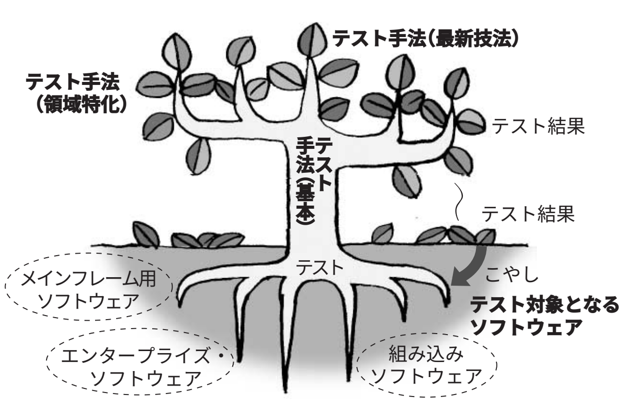


図4　テストを樹木でたとえると…
テスト手法の基本が樹木の幹，ソフトウェアが大地とすると，テストは根のようなものである．さまざまなソフトウェアをテストした結果を共通の経験として体系化したものが，テスト手法の基本になる．

ソフトウェア開発者のストレスがたまり，「どうして，こんな時期にトラブルを見つけちゃうの！」と，テスト技術者に八つ当たりする場面に出会うことがあります．もちろん，不ぐあいはソフトウェア開発者の問題であり，一時の感情の乱れなのですが…．そのようなときにも，冷静に，事実を淡々と報告し，ソフトウェア開発者の状況も理解したうえで対処できることが望まれます．

● 品質を維持しながらテスト項目を減らす

古典的なソフトウェア・テストには，「フルパス，フルスケールでテストする」という考えかたがありますが，最近の組み込みソフトウェアにこの考えかたを適用すると，品質を確立するために数兆～数十兆のテスト項目を消化しなければなりません．これほどのテスト項目を開発期間中にこなせるわけがありません．そこで，品質を維持しながら，テスト項目数を減らす方法が必要になります．例えば，境界値分析法や直行表の考えかたは，テスト項目数の削減を目的にしているといっても過言ではないでしょう．

これらの方法を実践することにより，開発中の品質は確かに向上しますが，不ぐあいは抽出しきれません．結局，テスト項目から外れた組み合わせが問題を起こすのです．抽出できなかった不ぐあいに対しては，「発生した」という事実に基づく経験を開発にフィードバックする再発防止活動によって対処していきます．このように，ソフトウェア・テスト技術はまだ未成熟であることを認識しながら，ソフトウェア技術者はくふうを続けなければなりません．

● 未成熟ゆえに，知恵を結集して進歩させるべき分野

ソフトウェア・テスト技術はまだまだ未成熟です．つまり，これからこの分野にかかわってしごとをする人たちは知恵を結集し，進歩させるべき分野であることを認識するべきです．昨今のテスト項目の爆発的増大や開発期間短縮の要求にこたえるには，ソフトウェア開発者がテスト項目を分析し，最終テストまでの品質確立工程全体をチェックしなければなりません．「品質の作り込み」と「テスト」のバランスをとりながら，最適なタイミングで効果的なテストを実施する必要があります．

世の中の役に立つ組み込みシステムを高い品質で早期に社会に提供することに情熱を燃やすのが，ソフトウェア・テスト技術者です．最高のテストを行うには，みずからの創意とくふうが重要です．日々，最新のテスト技術に耳を傾け，実践しながら研さんを積み，ソフトウェア・テストのプロフェッショナルを目ざしましょう．

参考・引用*文献

(1) Karl E. Wiegers；ソフトウェア要求 ―顧客が望むシステムとは―，日経BPソフトプレス，2003年7月．

すぎうら・ひでき　富士ゼロックス(株)

2

私の推薦書

「ライト，ついてますか ― 問題発見の人間学」



Donald C. Gause，G. M. Weinberg著
木村 泉訳
共立出版 刊，ISBN4-320-02368-4，21cm×15cm，161ページ，2,100円，1987年10月

ソフトウェア・テスト技術者はどうしてテストが必要なのかということを知るうえで，「問題は何なのか？」，「問題を抱えているのはだれか？」ということを，いつでも考えるくせをつけておくべきです．そのうえで，「問題とは，望まれた事がらと認識された事がらの間の相違である」という真理を冷静に受け止めなければなりません．

ソフトウェア・テストのシチュエーションで考えたとき，テストで起った問題は不ぐあいとして扱いますが，「不ぐあい＝問題」であるとしたら，それはだれの問題なのでしょうか？ 本書の考えかたに従うと，より幅広く問題をとらえることができるでしょう．テスト対象のソフトウェアの問題か，ハードウェアの問題か，要求分析の問題か，システム設計の問題か，さらにはテスト仕様の問題か．当事者についてもしかり．ソフトウェア開発者か，ハードウェア開発者か，システム開発者か，ソフトウェア・テスト技術者か，プロジェクト・マネージャか．

タイトルにもなっている「ライト，ついてますか」の話では，いかにまちがいに気づかせるかという話題が提供されます．テストのシチュエーションでは，問題が起こったときにそれをどう気づかせるかが一つのテーマとなります．つまり，テスト・オペレータが問題と認識しなかったら，テストは消化されるだけで，不ぐあいを検出できないわけです．問題を問題として気づかせるには，人の心理や行動にいかに着目するかということを考慮しないわけにはいきません．

ソフトウェア・テスト技術に特化した専門書はたくさんありますが，各論に重きを置いたものがほとんどです．一方，本書は，テストの専門書を読む前に問題の本質やそれをどう考えるべきかということに気づかせてくれる名著です．

05 マイコン開発者

M.P.I.

マイコン開発者とは

マイコン開発者とは，マイクロプロセッサやシングル・チップ・マイクロコントローラなどのLSIを開発・設計し，それを実用に供するまでの業務にかかわる技術者である．多分野にわたる専門性が必要とされる点から，マイコンはもっとも裾野が広い半導体デバイスと言える．また，マイコン開発者には，マイコンを使う側の視点も求められる．

● 多様な「プロ」が役割を分担して開発

マイコンのデバイス開発に携わるエンジニアはいくつかの職種に分類でき，それぞれ必要とする知識やスキルが異なります．ある意味，なにかしらの「プロ」と言える知識があれば，それなりの役割を果たせるのがこの商売の良いところです．ここでは代表的な職種のスキルを紹介します．

アーキテクトはマイコン開発の花形と言える職種で，基本的な設計仕様を決める技術者です．アーキテクトがひとりいれば，50人前のしごとができるというくらい重要な職種ですが，実はCPUコアを新規に設計するチャンスは，それほど多くありません．半導体の知識はもちろんですが，コンピュータについての造けい（単なる知識以上のもの），そしてアプリケーションについての理解が求められます．その中には，コンパイラやOS，ミドルウェアなど，ソフトウェアに関する知識も含まれます．

論理設計者には，RTL（register transfer level）設計などの論理回路の設計スキルは必須です．また，アセンブリ言語やC言語などにもなじんでいないと，マイコン分野では活躍の場が狭くなります．マイクロプロセッサともなると，相当に規模が大きく，構想力が要求されます．だれかから仕様書をもらわないとしごとができないというスキル・レベルでは，部分システムのそのまたパーツ・レベルの設計しかまかせてもらえず，おもしろいしごとはできません．

マイクロプロセッサはキャッシュなどの高速メモリのかたまりです．そういう意味で，トランジスタ・レベルを取り扱う回路設計のプロフェッショナルが必要とされる場面も少なくありません．メモリに加えて，集積されたマイクロコントローラには，A-Dコンバータなどのアナログ回路も混載されます．そのため，アナログ回路設計者も必要とされます．論理設計で行き詰まった箇所を，回路設計者のスペシャル・テクニックで突破するというケースもあります．

ここまではデバイス開発の関係者ですが，これだけではマイコンを市場に送り出すことはできません．以下のようなエンジニアも必要となります．従事者の人数から言えば，こちらのほうが多いかもしれません．ただし，デバイス開発はマイコン・メーカの内部で業務が完結することが多いのですが，以下の部分はサード・パーティと協業してあたることが多いようです．また，このような職種では，作る側と使う側がメンバ的にオーバラップしてきます．

まず，コンパイラやアセンブラ，デバッガなどのソフトウェア開発ツールの開発者，そして各種のミドルウェアの開発者やOSの開発者が必要です．最近ではマイクロプロセッサやマイクロコントローラを開発すると同時に，ターゲットとなるアプリケーションに近いところもいっしょに開発することがあります．そのため，アプリケーション寄りのソフトウェア・エンジニアの需要が増えています．

また，インサーキット・エミュレータ（ICE），あるいは上記のアプリケーション・ソフトウェアを開発するための開発ボードなどの開発者やマニュアル類を作成するテクニカル・ライタも求められています．

そして最後に，プロジェクト・マネージャが必要です．多様な社内外の職種のメンバをとりまとめてプロジェクトを進めなければならないので，多分野のすべてに通じているだけでなく，統率力が要求されます．また，多くのリソ

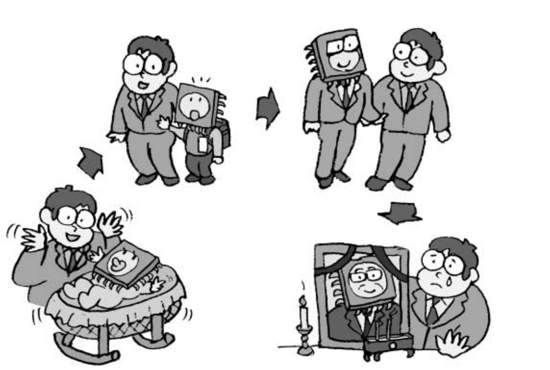


図5　アプリケーションの「ゆりかごから墓場まで」

マイコン開発はシステム開発とともに歩んでいく気の長い商売．量産開始以降も，顧客に対するサポートはえんえんと続く．場合によっては，アプリケーションの立ち上げから製品サイクルの終えんまで付き合うこともある．

ースを必要とするしごとなので，社内やプロジェクト・チーム内にもいろいろとあつれきが生じます．そのあつれきに対抗して，内ではプロジェクトを守り，外では競合メーカとやり合うだけの闘志が必要とされます．

● 部品事業だが，アプリケーションが見える

ほかのLSIと比べると，マイコン事業は足の長い商売です．CPUコアの開発に時間がかかるだけでなく，一度作ればかなり長期にわたってそのコアを使い続けます．また，顧客の側でも，新規にシステムを立ち上げるとき，その開発初期にどのマイコンを採用するかによって，さまざまなことが決まってしまいます．ソフトウェアへの投資も大きく，資産は再利用され続けます．

ほかの汎用部品などが設計の最後，すなわち調達に近い段階になってから選定・変更されうることを考えると，マイコン開発はシステム開発とともに歩んでいく気の長い商売であると言えます．そしてソフトウェアがあるため，量産開始以降も，顧客に対するサポートはえんえんと続きます．場合によっては，アプリケーションの立ち上げから製品サイクルの終えんまで付き合うこともあります(**図5**)．

これは苦労ではありますが，実際のアプリケーションが見えて，手ごたえを感じられる点で，やりがいも大きいと言えます．何に使われているかがよくわからず，とにかく言われたものを作って売るだけになりがちな部品事業のなかでは例外的なしごととも言えます．

● マイコンを使うのが楽しければ，資格あり

ソフトウェアでもハードウェアでもよいので，まずは使う側の立場で「楽しく」マイコンと付き合えるようであれば，マイコン開発に参加する資格はあります．その後，ソフトウェアなのかハードウェアなのか，半導体の中身の開発を志向しているのかシステム構築を志向しているのか，あるいはディジタルかアナログかなどを考えて職種を選び，プロフェッショナルになっていけばよいと思います．間口が広いだけ，活躍の場はきっとあります．

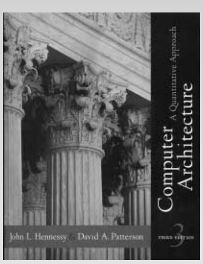
ただし，欲を言えば，「恐れを知らない男たち（もしくは女たち）」であってほしいと思います．変化は早く，つねにチャレンジしていく気持ちがないと，このしごとは続かないかもしれません．

M.P.I.　マイコン開発マネージャ

2

私の推薦書

“Computer Architecture: A Quantitative Approach”



J. L. Hennessy，D. A. Patterson 著
Morgan Kaufmann Publishers刊，
ISBN1-55860-724-2，23.5cm×18.7cm，
1,136ページ，89.95ドル，2002年5月（第3版），
※ 初版の刊行は1990年

少し古い本になりますが，J. L. Hennessy，D. A. Patterson共著の“Computer Architecture：A Quantitative Approach”は読んでおいたほうがよいでしょう．これは「読んでいないとモグリ」と言われる定番本です．MIPSとSPARCの生みの親がRISC台頭の勢いのあるときに出した書籍で，「教科書」として書かれているにもかかわらず，そのときの熱気というか，パワーが行間に感じられます．今でもこの本に書かれている考えかたにまちがいはないと思われます．そのためか，現在でも大学などの計算機アーキテクチャの定番教科書になっているようです．

本書が画期的だったのは，“A Quantitative Approach”と言っているように，「アーキテクチャを決めるときはきちんと得失を定量的に考えて決断しようよ」という，今となってはしごく当然の考えかたを提示していたからです．そのため，各章とも設計するときに必然的に直面することになる「トレードオフ」をどう定量的に判断して決めていくか，という視点で貫かれています．計算機アーキテクチャに限らず，トレードオフ評価において良い判断ができるのが良いエンジニアであるとすれば，本書はもっと広範囲の分野で読まれてもよい書籍であると言えます．各章末に，トレードオフ評価の訓練を行うのによい練習問題が多数あるので，これを解いて，数人で議論し合うと理解が深まるかもしれません．

ところで，なぜ定量的なトレードオフ評価が画期的だったのかを理解することは，CPUアーキテクチャをアートととらえていた古い時代を知らないと難しいかもしれません．そのあたりの背景まで理解して本書を読んでいくには，ときどき引き合いに出されている，かつて一世を風靡(ふうび)したVAX-11やIBM360などについての知識もあったほうがよいと思います．また，本書が出た後，CPUアーキテクチャのトレンドはスーパスカラ，VLIW（very long instruction word）を経て，マルチスレッド，SMP（symmetrical multiprocessing）へと流れています．そういうものに興味がある場合にも，本書で基礎的な考えかたを身に付けたうえで，そちらへ進むとよいと思います．

この書籍は，教科書としては大学院在学者程度を想定しています．計算機の算法など，基礎知識のない入門者には，姉妹書で，やはりPatterson，Hennessy共著の“Computer Organization and Design：The Hardware/Software Interface”を読むとよいかもしれません．どちらも翻訳本が出ていますが，先のことを考えると，英語の勉強をかねて原書で読むのがお勧めです．

06 ASIC/FPGA設計者

松本 仁

ASIC/FPGA設計者とは

ASIC（application specific integrated circuit）/FPGA（field programmable gate array）設計者とは，情報家電や産業用機器など，各種システムに搭載する特定用途向けLSIの企画立案から論理設計までを行うディジタルLSI設計者である．こうしたデバイスはおもに，機器メーカが企画して開発する．なお，カスタムLSIであるASICの場合，後工程のレイアウト設計（フィジカル設計）やアナログ回路設計は半導体メーカが担当する．

● アプリケーションや周辺LSIについての知識が必要

論理設計（ディジタル回路設計）を行ううえで最低限必要な知識は，道具であるEDAツールの使いかたと，回路の機能を表現するHDL（hardware description language）をマスタすることです．ただし，「HDLで記述できる」と言った場合，その中身が問題となります．ASIC/FPGA設計者は，論理合成によって，記述したHDLからどのような回路が生成されるのかを理解していなければなりません．

また，ASICやFPGAに接続される周辺LSIの機能や規格を理解しておくことがたいせつです．一つのシステムは，ASICやFPGAだけでなく，さまざまなデバイスから構成されています．したがって，ASICやFPGAに接続される部品の機能やタイミング，設計制約などをデータシートから読み取るスキルが必要になります．近年，ASICやFPGAの中にCPUコアや各種IP（intellectual property）コア，PLL，メモリ・インターフェースなど，さまざまな機能を盛り込み，付加価値の高いデバイスを構築することが盛んに行われています．データシートの内容をきちんと理解していないと，これらを適切に使用することはできません．

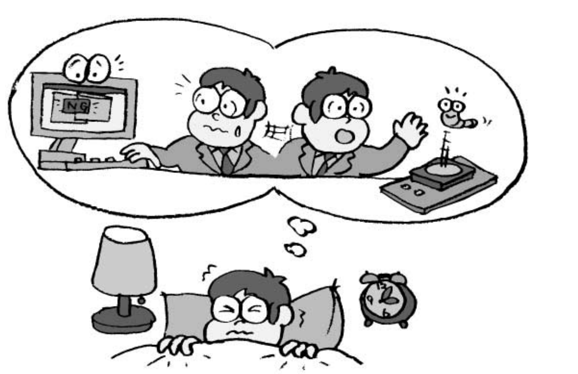


図6　夢の中でも「論理設計」

ASIC設計はプレッシャとの戦い．納期などで追いつめられていると，寝ているときも，設計業務で苦しんでいる夢をよく見る．

さらに，ASICやFPGAを設計するうえでは，最終製品についての理解が不可欠です．これは，システムの機能の大部分をASICやFPGAで実現することが多いためです．アプリケーション特有の機能や規格を理解していないと，システム仕様からASICやFPGAに搭載する機能を適切に切り出せません．したがって，自分が担当する製品に関する技術書や規格書の内容を正しく把握することが重要です．

加えて，今日の論理設計者は，単にHDLで機能を記述・検証すればASICやFPGAができ上がるわけではないことを理解しておく必要があります．例えばレイアウト設計の制約を考慮しながら論理設計の段階でフロアプランを行います．クロック設計やテスト設計などの知識も必要です．

● 設計品質の向上に神経を使う

ここでASIC設計とFPGA設計の違いについて，少し触れておきます．論理回路を設計するという点ではどちらも同じです．しかし，設計品質の考えかたがやや異なります．

FPGAは，短納期，安価な開発コスト，書き換え可能（すなわち機能の修正が容易）という安心感から，実機を動かして検出した不ぐあいを修正しながら完成度を高めることが多いようです．そのため，コーナ・ケースの検証もれなどで不ぐあいが出荷間際に見つかり，その対策に多くの時間を費やしているように見えます．

一方，ASIC開発では，0.18μmプロセスで数千万円，90nmでは数億円の開発費（NRE：non-recurring expenses）が必要となります．万一，仕様設計や論理設計にミスがあると，膨大な損失が発生します．また，競争の厳しい製品の市場投入時期が遅延すると，会社の事業の存続を揺るがす事態にも発展しかねません．そのため，審査やレビュー，設計品質の測定方法が各社ごとにルール化され，厳格に運用されています．

最近では1チップに集積できる回路の規模が増大し，設

計者ひとりですべての工程を見通せるケースは減少しています．実際，大規模ASICの開発には100名を超える設計者が参加します．デザイン・ハウスなどから設計者の派遣を受けて作業を進めるケースも多く，設計者間の意思の疎通をうまく行わなければなりません．そのため，各種仕様書のレビューや進捗管理，不ぐあいの修正管理を行って問題を早期に発見したり，早期に対策を打つことが重要になっています．また，ソフトウェア開発で実績のある手法(ソフトウェア・テスト技法など)をASIC設計に取り入れて，チーム全体の作業効率を引き上げることが増えています．

筆者はASIC/FPGA設計に携わって十数年になりますが，「仕様変更のないASIC/FPGA設計はない」というぐらい，要求仕様の変更に起因する再設計が頻繁に発生します．そのたびに機能仕様設計書の作り直し，再レビュー，回路の修正，検証という一連のフローを実施することになります．仕様変更の発生が後の工程になるほど，工程遅延のインパクトは大きくなります．早期に仕様を確定させることが理想なのですが，競合企業の動向など，社内外のさまざまな要因で仕様の確定が遅れるケースが増えています．

立場によって異なるとは思いますが，自分の設計したASIC/FPGAが最終製品に搭載され，エンド・ユーザの役に立っている場面に出会うと，しごとのやりがいを感じます．初めて電源を投入し，信号が正常に出ていることを確認すると，ホッとします．とくに再設計が開発コストに大きなインパクトを与えるASIC設計は，プレッシャとの戦いです．納期などで追いつめられていると，寝ているときも，設計業務で苦しんでいる夢をよく見ます(**図6**)．

● 電気・電子系以外の出身でもハンデは少ない

筆者らの部署を見渡すと，ASIC/FPGA設計者の学生時代の専攻は多種多様です．論理設計のしごとに限れば，かならずしも電気・電子系の出身者が有利であるとは言えないようです．それよりも，システムの機能を一つ一つ整理し，最適な機能ブロックに分割する能力のほうが重要です．また，標準方式の勧告，および規格書に記載されている数式やアルゴリズムなどを回路に落とす場合には，数式やアルゴリズムが持っている規則性などを見抜き，コンパクトでシンプルな回路を構築する能力が求められます．最近では有益な参考書が多く出回っており，これからこの分野に進もうという人にとってもハンデは少ないと思います．

ただしASICなどは，一度プロジェクトがスタートすると短いもので半年，長いものでは1年半程度の開発期間を要します．そのため，設計者のモチベーションをいかにして高く保つかが大きな課題となってきています．

まつもと・ひとし
三菱電機(株) コミュニケーション・ネットワーク製作所

2

私の推薦書

「映像メディアの世紀 — ビデオ・男たちの産業史」



佐藤正明 著
日経BP社 刊，
ISBN4-8222-4159-9，
19cm×13cm，646ページ，
1,995円，1999年10月

本書は，NHKの「プロジェクトX」にも取りあげられた，VTR(video tape recorder)開発やその後の8mmビデオ開発をめぐる産業史について書かれたノンフィクションです．官庁まで巻き込んだ二つの規格(VHSとベータ)をめぐるファミリ作りの舞台裏まで記載されており，20世紀最大の家電製品であるVTRの開発者の苦労が読み取れます．

その中で，"ミスターVHS"こと，日本ビクターの高野氏の「VHSを世界規格に育てたい」という夢や，「いったん決めた規格は，ユーザがいる限り，たとえ会社がつぶれても守り抜くべきだ」という信念など，技術者魂を揺さぶられる箇所の多い内容でした．

いつの時代でも，市場からの声に耳を傾けることは重要です．VHSの開発では，家庭用VTRのニーズの洗い出しに2年の歳月をかけ，マトリックスにまとめあげたのだそうです．この手法は開発の基本として現在も踏襲されています．変化の激しい現在でもこの手法は開発の本質であり，私たちエンジニアが忙しさで忘れている点でもあります．

一方，ソニーはクリスマス商戦用の広告でVTRをPRしました．当時の人気テレビ・ドラマである「刑事コロンボ」と「刑事コジャック」は同じ曜日の同じ時間に放映されており，同時に見ることができませんでした．しかし，タイム・シフト・マシンである家庭用VTRに録画すれば両方を見られるとPRしたのです．そして，これが発端となり，ハリウッド(米国映画界)との著作権争いが勃発し，最後にはソニーが勝利して今に至っています．

今から考えるとあたりまえなのですが，映画館での公開が終了した後，家庭で映画を手軽に鑑賞できるようになりました．これにより，ハリウッドはさらなる利益を得，ますます発展し続けています．文化まで変えたという点で，VTRは単なる家電製品の枠を超えました．20世紀における，後世に誇れるもっとも成功した電子機器の一つであることは議論の余地がありません．

07　LSIパッケージ設計者

中島宏文

LSIパッケージ設計者とは

LSIパッケージ設計者とは，顧客である電子機器メーカの実装面の要望を満足し，LSIチップの性能をフルに引き出すパッケージを提案し，具現化していく技術者である．設計者ということばから，CADツールに向かって黙々と作図している姿を連想するかもしれないが，それは全体の業務の3割程度である．高密度実装を目指す顧客とともに最良のパッケージを検討したり，基板メーカと配線パターンの打ち合わせを行ったり，組み立て技術者と設計基準の緩和計画を立案するなど，コーディネートのために多くの時間を費やしている．

● 電気のほかに，熱や信頼性，実装などの知識が必要

LSIパッケージ設計者に必要な知識は，電磁気学，熱学，力学，材料物性，信頼性技術，実装技術などです．半導体デバイス（チップ）の電極をワイヤやバンプを介してパッケージの外部端子に結線するのですが，その経路上で電磁現象に起因するノイズが発生し，信号の伝播が阻害されます．電磁気学に基づいてノイズの影響を定量的に計算することは，大学で電磁気学を専攻した人であっても難しいものですが，パッケージ設計者はこうした理論を定性的に理解し，ノイズの発生を予測できるだけの知識と経験が必要です．

さらに，チップが消費する電力によって発生する熱をパッケージから放熱させる熱設計は，電気設計とともに重要です．チップの微細化に伴って，チップの温度上昇がリーク電流の急増を引き起こし，デバイスの破壊につながることが指摘されています．

半導体デバイスの信頼性保証の観点からは，実装リフロ時のパッケージの反りやチップの抗折強度，基板とパッケージの熱膨張差による応力などを把握する必要があります．パッケージの信頼性は応力や耐蝕性，および不純物との戦いです．そのため，パッケージの材料物性を理解して材料の選択・開発を行う必要があり，これらの知識のうえに半導体デバイスの信頼性が確保されているのです．

筆者らの会社でも，上に述べた知識をまとめたテキストを作成したことがあります．しかし，若いエンジニアからの反響は今ひとつでした．そのときに思ったのは，やはり知識は与えられて手にするものではなく，自分から取りにいくものである，ということです．情報が氾らんしている現在，欲しい情報やその手がかりはたいていインターネットから得られます．必要なのは，欲しいという意欲と検索する行動力です．



図7　顧客との会話では相手企業との関係に配慮
LSIパッケージ設計者は顧客（電子機器メーカ）と直接対話する機会が多い．ときには，相手企業との関係に配慮したことばの選択が求められることもある．

● 顧客と直接対話する機会が多い

LSIパッケージ設計者は顧客（電子機器メーカ）と直接対話する機会が多く，自社の技術を紹介し，顧客の要望に沿ったLSIパッケージを提案します．最近では，一つのパッケージに複数のチップ（ダイ）を組み込むシステム・イン・パッケージ（SIP：system in package）に興味を持つ顧客が増えています．こうした顧客は極限の高密度実装を目指している場合が多く，「現在の技術の範囲でもっとも小型で薄くするにはどうすればよいのか」とよく聞かれます．そして，「当然，実装信頼性は保証してくれるんだろうね」，「仕様に記載されていなくても，問題が起こったら，『知らない』なんて言わないだろうね」とたたみかけてきます．しかし，面と向かって，「仕様に入っていないのですから，問題が起こったらお客様の責任です」とは言いにくいものです．生産工程の能力と過去の経験をもとに，「そのときはお互いに協力して解決しましょう」と言うのか，「実力値からみて自信がありません」と言うのか，相手企業との関係に配慮したことばの選択が求められます（**図7**）．

パッケージ設計者は顧客との窓口でもあるので，ときには社内の役割を越えた責任も負わされます．筆者らが端子ピッチの非常に微細なFBGA（fine pitch ball grid array）を開発したときのことです．パッケージとテスト・ソケットの開発準備がばらばらに行われており，顧客から見て供給側の体制に不安を覚えたのでしょう．パッケージとテスト・ソケットの開発状況を一括して報告することを求められました．そのため，顧客との打ち合わせの前に社内の開発状況の進捗をチェックせざるをえなくなりました．しかし，このおかげで以降は各部署の進捗の整合がとれ，開発が円滑に進みました．パッケージ基板の設計や各種製造図面の作成などは，こうした顧客とのやりとりの合間に行う，誤りの許されないしごとです．

社内では，チップのレイアウト設計の段階から打ち合わせに参加して，パッケージ基板の配線がもっともシンプルなパターンになるように，チップの縦横比やパッド配置などを協議します．最小の形状とシンプルな配線こそがパッケージ・コストを抑制し，利益を生み出す源泉となります．

LSIパッケージ設計者は設計・試作後に行う設計審査と，信頼性評価後に行う量産移管審査をパスして，初めて開発したパッケージを生産技術部門に引き渡すことができます．そのため，試作評価結果のデータや工程能力，材料物性などを詳細に把握して審査に臨みます．量産移管審査会では，あらゆる可能性を検証する品質管理技術者の質問（ときには意地悪に思える質問もあるが…）に回答してリスク・レベルを明示することが要求されます．そして，不足のデータなどがあれば再審査になります．

● 業務だけでなく，個人レベルの地道な勉強も必要

知識やスキルについては，業務の中で身に付けるものと，個人で心がけて学習するものがあります．

前者の例ですが，例えばLSIチップのレイアウト設計者にパッド・レイアウトに対する要望を説明していたとき，チップの設計手順やツールの話になったとします．このとき，自分の担当分野ではないと考えて知識の習得をそこで放棄するのか，それとも「門外漢ですみませんが…」と断って，レイアウト設計者に質問する態度を身に付けるのかによって，数年後の業務知識に大きな差が出ます．自分の担当分野を中心として，周囲に知識の幅を広げていく"V字型"の知識習得を心がけていただきたいと思います．

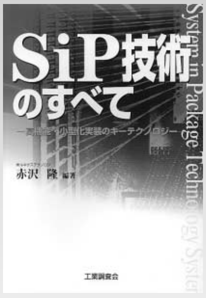
後者については，例えばシステム・イン・パッケージの開発において，チップ間のデータのやりとりに高速シリアル伝送が使われる状況を想像してください．高速シリアル伝送では，ノイズの知識なしにパッケージ基板を設計できません．ノイズに配慮していなかったために，顧客の実機に組み込んだときに動作しないということが起こりえます．このようなことにならないために，日常業務とは別に，ノイズや伝送線路理論について，個人で地道に勉強することがたいせつです．

なかじま・ひろふみ　NECエレクトロニクス（株）

2

私の推薦書

「SiP技術のすべて」



赤沢 隆 編著
工業調査会 刊，
ISBN4-7693-1242-3，21cm×15cm，
244ページ，2,625円，2005年7月

本書は2005年7月に出版されたパッケージ関連の最新解説書です．システム・イン・パッケージのシステム設計技術やパッケージ技術について，各分野の7人のエキスパートがそれぞれ得意な分野を分担して執筆しています．パッケージ技術を含むシステム・イン・パッケージの最新状況がていねいに解説されており，パッケージ技術者の入門書としてお勧めです．

とくに第5章の「SiP設計技術」，第6章の「SiP構造技術」，第7章の「SiP実装技術」では，システム・イン・パッケージの基本となるパッケージ技術をもとに，最新技術がわかりやすく解説されています．また，基板層数を見積もるツールの例として，NEC情報システムズの「GENISSNX PKG-Estimator」が紹介されています．商談開始時点でシステム・イン・パッケージのコストを見積もる難しさと重要性が端的に述べられており，感銘を受けました．

少し残念に感じたのは，まとめとなる第9章の「将来のSiP」です．本書全体がシステム・イン・パッケージのプラスの面に焦点を当てているためか，この章は単なるシステム・イン・パッケージのカテゴリ分類ととられかねない図で終わっています．将来の展望というほど時間軸が感じられませんでした．長年パッケージ分野にかかわってきたエキスパートの方々が時間をかけて執筆されたのですから，最後はパッケージ技術の将来に対する洞察のことばが掲載されてもよかったのではないでしょうか．

とはいえ，本書はシステム・イン・パッケージを一般の人々に理解してもらうという役割を十二分に果たしています．日常業務の間や自宅で，休日もいとわず時間を割いて執筆された著者の方々に脱帽しました．

08 プリント基板設計者

嶋田茂晴

プリント基板設計者とは

産業用機器や家電製品，自動車などの機能は，電子回路によって構成・制御されている．そして電子回路は，電子部品間をプリント基板上の配線パターンで接続することによって実現されている．プリント基板は製品ごとに異なる機能や回路，形状（デザイン）を実現するため，すべてカスタムメイドになる．プリント基板設計者は，回路図をもとに，プリント基板の仕様や部品実装などを考慮しながら，部品配置，およびパターンの設計（アートワーク）を行う．

● 回路知識に加えて製造方法や規格・規制の知識が必要

プリント基板設計者は，回路図上の抵抗やコンデンサ，トランジスタなど，電子部品を示す回路記号はもちろんのこと，電子部品としての機能や役割，それらを組み合わせた基本回路のしくみ，および動作を理解していなければなりません．また，電源回路や発振回路，バス・ライン，デカップリング・コンデンサなど，動作やノイズに影響する回路の知識を持っていることが望ましいケースもあります．そして，電子部品のカタログや基板外形図などに記載されている寸法を理解するためには，図面を読む基本的な知識が必要です．

製造技術の進歩とともに，プリント配線板の種類や工法，特徴，製造仕様は複雑になっています．これらの基本的な事項を理解しておく必要があります．プリント配線板の種類や工法はコストに影響します．とくにビルドアップ・プリント配線板の場合，構造が同じでも製造会社によって設計・製造仕様が異なることが多く，基板の設計に着手する際にはきちんとメーカに確認する必要があります．

設計にはプリント基板設計専用のCADツールを利用します．そのため，基本的なコンピュータ（パソコンが主流）やツール操作の知識が必要になります．

電子機器を取り巻く法律には，電源などの高圧回路規格，ノイズ規格，安全規格などがあります．これらの内容も理解しておきましょう．

最近では，プリント配線板も電子部品の一つとして，環境対応を求められるようになってきています．そのため，素材やレジスト，表面処理などの知識が必要になります．環境対応素材の場合，機械加工の条件が従来の素材と異なることがあります．プリント配線板の設計・製造仕様が変わることも多いので，きちんと確認してください．



図8　将来の製品や開発の背景がいち早くわかる

顧客の信頼を獲得すると，次期製品の開発のときに指名を受けるようになる．将来の製品動向や開発の背景をいち早く知ることができる．

● いちばん最後に開発が始まり，最初に必要となる部品

プリント基板は，電子機器の形状や機能，機構，電気回路，必要な電子部品など，基本的な事項が決定した後，カスタム部品として設計・製造に入ります．ところが製品を組み立てる際には，いちばん最初に必要な電子部品であるため，つねに短期間の開発が求められます．

設計途中の仕様変更に対応しながら，電気特性や製造性の要求も満足する必要があり，「技術力＋体力」で勝負することになります．どんなに優秀な技術者であっても，ひとりですべての業務をこなすことは不可能です．そのため，つねにチーム内のコミュニケーションを取り，情報を共有しながらしごとを進めます．

このしごとのおもしろい点は，相手（顧客）も技術者なので，駆け引きなしに相談や提案ができることです．製品開発では，何度かの試作を繰り返しながら，完成度を高めていくことになります．そのときどきの問題点を解決するため，提案を行うことも重要なしごとです．また，つねに相談を受ける立場になると，自分が頼りにされているという実感もわきます．

自分が設計したプリント基板が製品に組み込まれて世の

中に出たときは，開発に携わったメンバのひとりとして，達成感や満足感を味わえます．これは，技術者としての喜びや自信につながります．顧客との連帯感や信頼も生まれ，次期製品の開発のときに指名を受けるようになります．こうなると，苦労が喜びに変わります．

将来の製品動向や開発の背景などをいち早く知りえる立場にある点も，得をした気分になります（**図8**，企業秘密は守ることが前提となるが…）．

● なくてはならない縁の下の力持ち

筆者は多くのプリント基板設計を行ってきましたが，検討のための試作を省いて最初から量産用の設計を行い，通常よりも短い期間で開発できた経験があります．また，自分がかかわった製品の開発秘話が会社のWebサイトに掲載されたこともあります．新しい製品のプリント基板設計が完了し，自分なりにチェックして，回路設計者へ検図を依頼した結果，問題点の指摘を一つも受けなかった経験が一度だけあります．これは，今でも自分の勲章として誇りに思っています．おそらく今後も経験できないことだと思います．

プリント基板は，脚光を浴びることの少ない，縁の下の力持ち的な存在ですが，なくてはならない重要な電子部品です．同じ回路，同じ条件が与えられても，設計者が異なると，プリント基板の設計結果は異なります．当然のことながら，正解は一つではないので，設計者の力量を示せる職業です．

携帯電話やパソコン，自動車，医療機器など，プリント基板はあらゆる製品に組み込まれており，生活の進歩にもひと役買っています．一見地味なしごとに見えますが，やりがいや喜びは大きく，失敗しても，それは次につながる重要な経験となります．

最先端の製品に採用されるプリント基板の設計は，日本人のち密さ，器用さが生きる職業であると筆者は考えています．

しまだ・しげはる
(株)シイエムケイ回路設計センター

2

私の推薦書

「ぷりんとばんじゅくⅡ 新入社員のための プリント配線板入門ー設計編ー」



田中弘文著
社団法人日本プリント回路工業会 刊，
A5判，97ページ，2,100円[注]，
2003年6月（第4版）

プリント基板設計の知識を得るための参考書として，社団法人日本プリント回路工業会（JPCA）より刊行されている書籍「ぷりんとばんじゅくⅡ 新入社員のためのプリント配線板入門－設計編－」を紹介します．電子部品や電子回路の詳細な内容はほかの参考書に譲るとして，本書ではプリント基板設計の基本となる知識を会得でき，具体例を知ることができます．

内容としては，プリント基板設計に対する基本的な考えかた，基板の種類，設計の具体的な手法，マニュアル設計とCAD設計，設計と製造の関係などについて解説しています．CADツールが普及した現在でも，プリント基板設計の基本は，回路図や部品表，部品カタログ，製造仕様などの資料に基づくマニュアル設計（人手による設計）だと筆者は考えます．プリント基板上の部品配置やパターン配線は，顧客や回路設計者から渡される回路図の流れや設計指示書の指示に従って行わなければなりません．こうした基本を身に付けるのに本書は役立つことでしょう．

「ぷりんとばんじゅくⅤ 新入社員のための実装入門」



榧場正男著
社団法人日本プリント回路工業会 刊，
A5判，99ページ，2,100円[注]，
2002年6月

もう1冊，実装という観点から，「ぷりんとばんじゅくⅤ 新入社員のための実装入門」が同じくJPCAから刊行されています．部品の種類や形状，接合（はんだ付け），プリント配線板の種類，および加工の観点から解説されています．コスト低減や品質確保，大量生産の視点で設計時に配慮するべき事がらが説明されています．さらには，製造や性能，コストなどのバランスのとれた良い設計を行う場合に必要な知識が得られる内容となっており，新人エンジニアのみなさんの参考になると思います．

注：日本プリント回路工業会の会員会社および会員会社の社員は1,050円で購入できる．

09　シミュレーション技術者

金子俊之

シミュレーション技術者とは

シミュレーション技術者はいろいろな技術領域で活躍しているが，ここではプリント基板の電気特性をアートワーク（レイアウト）設計段階，もしくはアートワーク設計前に，シミュレーション技術で確認・最適化するしごとについて述べる．シミュレーション技術者は，SPICEシミュレータまたは伝送線路解析ツールを使って基板の等価回路モデルを作成し，ドライバICやレシーバICなどの部品情報（IBISモデル，SPICEモデル）をもとに，伝送線路の電圧波形を解析する．

● ツールの限界とモデルの特性を理解する

近年，基板上を伝播する信号の周波数が上昇するにしたがって，基板のアートワーク設計が重要になってきています．とくに基板上のタイミング・マージンが減少し，基板のアートワーク設計がタイミングのよしあしに影響を与えるようになりました．また，高速信号の電圧レベルも低下し，電圧変動をできるだけ抑えられる最適なアートワークが求められています．そこで，シミュレーションを活用し，タイミングや電圧変動を最適化しながらアートワーク設計を行うようになってきています．

多くのシミュレーション技術者は，市販ツールを使って伝送線路解析を実施しています．市販ツールを使う場合，そのツールの限界をよく理解して，ツールに合った使用方法を選択することが必要になります（**図9**）．例えば，基板上の配線を等価回路モデルにしたとき，そのモデルに誘電損失や導体損失が含まれていないと，高周波の信号解析精度は悪くなります．しかし，配線長が短く，周波数がそれほど高くないのであれば，このモデルの解析時間は短くなり，有効な解析と言えます．このように，解析対象となる周波数や構造（例えば，複数の基板を接続した状態で解析するのか，システム・イン・パッケージのように半導体パッケージ上にある複数の半導体チップを接続した状態で解析するのか）によって，適切な解析ツールが異なってきます．また，市販ツールの多くはSPICEをベースとしています．SPICEの文法を理解することで，等価回路モデルやデバイス・モデルの内容を理解しやすくなります．



図9　ツールを知る，選ぶ
市販ツールを使う場合，そのツールの限界を熟知して，ツールに合った使用方法を選択することが必要になる．

基板上の配線は，伝送線路として等価回路モデルに置き換えられます．もっとも単純な等価回路は，R（抵抗）成分とL（インダクタンス）成分が直列につながり，GNDとの間のC（キャパシタンス）成分が並列につながったラダー（はしご段）回路です．等価回路モデルの作成方法は，使用するツールによって異なりますが，解析周波数範囲を理解して，最適なモデルを選択できるスキルが必要となります．そのためには，どのような周波数特性になるのかをモデルごとに理解しておく必要があります．例えば，先ほどの簡易的なRLCラダー回路の場合や，Sパラメータで記述されたタッチストーン（Touchstone）形式の等価回路モデルを使用する場合，ビアや配線コーナの等価回路モデルを組み込む場合などについて，解析精度と解析時間のトレードオフを検討する必要があります．

また，デバイス・モデルについても同様のことが言えます．一般にはIBISモデルがよく使われていますが，IBISモデルのフォーマットをよく理解して使うようにしましょう．

伝送線路シミュレーションが終わったら，その結果をもとにタイミング検証や電圧変動検証を実施し，伝送線路の信号波形を最適化します．最適化を行うおもな手法として，以下のものが挙げられます．

- 基板アートワーク（部品配置，配線パターン）の変更
- 回路設計の変更
- 組み込みソフトウェアの変更

これらのどの方法を採用するかは，プリント基板設計者，

回路設計者，組み込みソフトウェア開発者とよく議論する必要があります．このとき，コストや開発期間について考慮しながら，最適な提案を行います．シミュレーション技術者には，それぞれの変更によってコストや開発期間にどのような影響があるのかを把握するスキルが求められます．

● 多様な製品の開発に参加できる

基板の伝送線路シミュレーションでは，基板のアートワーク設計と連携し，製品の開発者（おもにシステム設計者）と連絡をとって作業を進めます．そのため，対象とする製品についての詳細な情報が入ってきます．伝送線路シミュレーションを行った設計が問題なく動作したということを聞くと，このしごとをしていて良かったと思います．

基板の伝送線路シミュレーションというのは，その製品の開発サイクル全体から見ると，ごく一部の工程にすぎません．しかし，製品の開発者が手間をかけずにシステムを動作させることができるようにアドバイスし，ときには半導体メーカと協力しながら，製品の開発を手助けするしごとは，とても魅力的です．とくに基板メーカや基板設計会社のシミュレーション技術者は，特定の製品だけを扱うわけではなく，多様な製品の開発にかかわることができます．

ただし，良いことばかりではありません．伝送線路シミュレーションの波形と実測波形が違うと言われることもあります（最近は少なくなってきたが…）．このときは，根気よくその原因を追求することになります．そして，その苦労は次のシミュレーションに生かされます．

また，製品の開発期間はあらかじめ確定しているので，シミュレーションにかけられる時間も決まってきます．その中で最適な結果を導き出すために苦労することもあります．

● 基板，パッケージ，チップを含む包括的な知識が必要に

基板の伝送線路シミュレーションは，今後も重要な技術であり続けると筆者は考えています．これに加えて，最近では，基板から出る電磁放射ノイズの解析（EMI）や電源電圧の変動を抑えるためのシミュレーションも注目されています．これらのシミュレーション技術についても，まずは伝送線路シミュレーションを理解したうえで勉強したほうが，理解が早いと思います．

また，基板上だけで信号波形や電源電圧を最適化することには限界があります．LSIパッケージやLSIチップなどを含めたシステム全体として最適化する必要性がでてきています．これまでは基板の知識だけあればよかったのですが，より広い知識が求められるようになってきています．

これからシミュレーション技術者を目ざそうと考えられている方は，基板や半導体といった個々の部品にとらわれず，幅広く通用する技術知識を持つように努力されることをお勧めします．

かねこ・としゆき
（株）トッパンNECサーキットソリューションズ

2

私の推薦書

「SPICEによる電子回路設計 －回路シミュレータPSpice入門」



John Keown 著
東京電機大学出版局 刊，
ISBN4-501-31550-4，
A5判，340ページ，
2,310円，1993年6月

本書は電子回路におけるSPICE解析事例を紹介したものですが，SPICEの文法を初歩から学ぶのに適しています．ディジタル信号を解析しようと考えている方でも，以下の内容は参考になります．

第1章は直流回路解析について書かれており，基本的な回路構成をSPICEで記述した例が示されています．この中には，キルヒホッフの法則の検証や，テブナンの定理の検証などがあります．筆算の結果と比較しながら，SPICEで解析した結果を確認できます．また，Fエレメント（電流依存性電流源）について詳細に解説されています．Fエレメントはクロストーク・ノイズ解析などを行う際の等価回路モデルとして使われるケースがあります．

第2章は交流回路解析について書かれています．ここでは，各種回路構成により，共振がどのように起こるのかが示されています．この内容を理解すれば，基板の信号配線や電源配線によって共振が起こる原理を理解できると思います．

第4章の「多段増幅器，周波数応答および帰還」では，ポール（極）が二つある帰還増幅器のモデルを例に，過渡解析時にオーバシュートが発生するメカニズムについて記述されています．これはディジタル回路の解析においても参考になります．さらに，第6章の「過渡応答と時間領域」は，増幅器のステップ応答について説明しています．ここではディジタル信号波形を解析したときに負荷の容量が変化すると信号波形にどのような影響があるのかを理解できます．

各章の終わりと巻末に各エレメントの記述方法がまとめられています．これらは，各種シミュレーション・ツールによって微妙に違うことがありますが，SPICEの文法を理解するうえでは参考になります．

10 RF技術者

市川裕一

RF技術者とは

RF技術者のRFは“radio frequency”の略である．一般に，RFはHF帯(high frequency，3MHz～30MHz)，VHF帯(very high frequency，30MHz～300MHz)，およびUHF帯(ultra high frequency，300MHz～3GHz)の一部(1GHz以下)を指すが，SHF帯(マイクロ波，super high frequency，3GHz～30GHz)も含めて広義で使われる場合もある．上記の周波数帯を取り扱う機器や回路，モジュールなどの開発・設計に携わるのがRF技術者である．そのしごとの範囲は非常に広く，回路設計はもちろん，RF回路のパターン設計(アートワーク)も基本的にRF技術者のしごとの範ちゅうである．さらに筐体(ケース)の設計を行うこともある．

● 交流理論のほか，ディジタルや電磁気学の知識も必要

RF回路は特殊な回路ではなく，周波数の高い信号を取り扱うアナログの交流回路です．そのため，電気回路で習う交流回路理論を理解していることが最低限の知識として必要です．RF回路でもダイオード，トランジスタ，FETなどの半導体素子を使うので，その動作や使いかたに対する知識も必要です．さらに，アンテナや分布定数回路を取り扱う場合には，電磁気学の知識も必要となります．

RF回路設計に必要なのは，上記の知識だけではありません．スイッチ回路と発振回路を例に，必要な知識を見てみましょう．



図10　忍耐強さと柔軟さ
RF回路の調整には忍耐強さが必要である．それと同時に，煮詰まってしまったときに発想を転換できる柔軟さも求められる．

まずスイッチ回路ですが，ディジタル信号でスイッチ素子を制御するのでディジタル回路の知識が必要です．また，スイッチの駆動回路をトランジスタやFETで構成することもあるので，トランジスタやFETのスイッチング回路の設計知識が必要になります．

次に発振回路は，PLL(phase-locked loop)回路と組み合わせて使われることが多いので，PLL回路を制御するためのディジタル回路の知識が必要になります．さらに，シングル・チップ・マイクロコントローラで制御することが多々あるので，マイクロコントローラのプログラミングもできるにこしたことはありません．そのほかに電源ラインの安定化やノイズ対策などの知識も必要になります．

このように，RF技術者にはディジタル回路をはじめ，さまざまな回路設計の知識やスキルが必要になります．これらの知識やスキルを一度に習得・吸収することはできません．さまざまな回路設計を行う中で習得・吸収するとともに，ふだんからの勉強も必要です．さらに電気的な知識に加えて，パターン設計や機構図面設計の知識も必要です．

RF回路の調整は非常に根気のいる作業です．ですから，忍耐強い人のほうがRF技術者に向いていると思います．それと同時に，煮詰まってしまったときに発想を転換できる柔軟さも必要だと思います(**図10**)．

● 文献や資料の収集から始め，1年後にようやく量産

RF技術のしごとに携わっていて苦労することはいろいろとありますが，その中でも大きなものを二つ挙げたいと思います．

一つ目は，RF回路設計に役立つ日本語の書籍や文献が少ないという点です．近年は，国内で出版されたRF技術関係の書籍も多く見かけられるようになりましたが，教科書的なものが多く，実設計に使える日本語の書籍はあまりありません．そのため，いやおうなしに海外の文献を取り寄せ，英語で書かれた書籍を読解する必要に迫られます．

二つ目は，シミュレータなどで設計したとおりに回路が動作してくれないということです．RF回路の場合，シミュレータによる設計完了は回路設計の完了ではなく，回路の基本設計の完了です．ここから実際に試作を行い，回路

の特性を引き出し，磨き上げていく作業が始まります．所望の特性が得られない場合には，特性が実現できるまで延々と調整・試行錯誤を繰り返すこともあります．これがRF回路設計でもっとも苦労する点です．しかしその分，苦労した回路の設計が完了したときには大きな達成感を味わうことができます．

苦労の多いRF回路設計ですが，自分が設計した製品や自分が設計した回路・モジュールが組み込まれた製品を目にしたとき，あるいはそれが問題なく機能して役立っていることを聞いたとき，その充実感が次の製品設計へ取り組む力となります．

筆者の中でいちばん強く印象に残っているしごとは，RF技術に携わってわずか1年目，まだ新入社員だったころにそのすべてを任されたモジュールの開発です．文献や資料の収集から始まり，約1年後にようやく量産にこぎ着けることができました．この製品の開発が，筆者にRF回路設計のたいへんさとおもしろさを教えてくれました．

● シミュレータを使っていろいろな回路を経験する

最近はRF技術者のしごとも細分化が進み，いったん配属が決まってしまうと，1種類の機器，回路，モジュールの開発・設計にずっと携わることになり，いろいろな回路の設計をあまり経験できません．そのような環境に置かれてしまった場合，シミュレータを使ってさまざまなRF回路の動作や設計に触れることをお勧めします．十分な機能を持つフリーのRF/マイクロ波シミュレータが簡単に入手できるので，やる気さえあれば自宅で勉強することも可能です．シミュレータを使えることが回路を設計できることではないので，書籍や文献による勉強もあわせて行う必要があります．

RF回路の場合，実際に回路を試作してみると，シミュレーションでは見られなかったさまざまな現象や不ぐあいに遭遇します．その原因や不ぐあい箇所を見つけ，迅速に対処するためには経験が重要になります．自分が置かれた環境の中で，できるだけたくさんの回路に触れ，みずからの手で調整を行ってください．

文献や書籍からの知識と実際の回路の経験を組み合わせることによって，RF技術者としてのレベルが向上していきます．残念ながら近道はありません．

効率的にさまざまな回路や周波数帯での設計経験を積むための手段として，転職するという方法もあります．筆者の場合，結果的にこの方法で設計経験を積むことになりました．

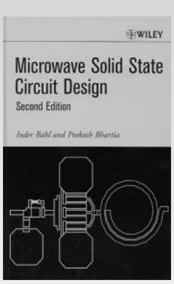
いちかわ・ゆういち
アイラボラトリー

2

私の推薦書

「Microwave Solid State Circuits Design, Second Edition」

Inder Bahl，Prakash Bhartia 著
Wiley-Interscience社 刊，ISBN0-471-20755-1，24.0cm × 16.0cm，906ページ，162.95ドル，2003年4月



国内で出版されている高周波・マイクロ波回路関係の書籍は，理論中心で教科書的なものが多く，設計の参考になる実用的なものは少ないように感じます．一方，海外の書籍は実用的なものが多く，その種類も非常に豊富で，選ぶのに苦労してしまうほどです．ここではその中から1冊の本を推薦します．

本書のタイトルには"Microwave"ということばが入っていますが，実際にはRFからマイクロ波まで広く対応しています．非常に幅広くさまざまな回路が解説されているので，本書を推薦しました．この本は15章で構成され，約900ページあります．

RF回路の設計で重要になる伝送線路の解説から始まり，カプラやフィルタなどの受動回路，能動素子（トランジスタ，FET）や受動素子（各種ダイオード），アンプなど，各種回路の解説が詳しく掲載されています．Second Editionになって，最新のRF MEMSの解説も追加されました．理論的な説明だけでなく，回路設計の計算例も掲載されています．

英語で書かれているというハードルはありますが，これから高周波・マイクロ波回路の勉強を始めようとしている方，そしてすでにこの分野の回路設計に携わっている方のどちらにも役立つ書籍だと思います．

完ぺきな書籍，これ1冊あれば十分という文献はまずありません．ここで紹介した書籍も実用的で良い本ですが，高周波・マイクロ波回路の設計に必要なことをすべて網羅しているわけではありません．本書も含めて複数の書籍を購入し，それぞれの本で内容が不足していると感じる部分をほかの本で補うことをお勧めします．

11 MEMS設計者

相馬伸一

MEMS設計者とは

MEMS(メムス)(micro electro mechanical systems)やセンサなどを開発するうえで，設計はたいへん重要な工程である．製品の特性や精度，信頼性はもちろん，製造方法や製品コストまでもが設計力に左右される．いわば，設計はMEMS製品のコンセプトそのものであり，その作業内容は多岐にわたる．MEMS設計者のしごとは，基本構造設計や素材の選定に始まる．昨今，さまざまな新材料がある中で，各素材の物性が製品特性を満足するかどうか，構造についてはその構造・設計精度が実現(製造)可能かどうかなどを吟味し，製品全体の取りまとめまでを行う．

● 材料や構造解析の知識が必要

MEMS設計者に要求される知識は，材料物性や材料力学といった材料に関する特性，シミュレーション技術，製造技術，コスト，量産性など，製品を世の中へ送り出すためのすべての項目が含まれます．なぜなら，MEMSの場合，設計の段階で製品のほぼすべてが決定してしまうためです．

設計者がMEMSの構造を設計する場合，目標仕様に合わせて最適な構造を検討していきます．その際，もっとも重要なものは材料の知識です．材料の選定は，その製品の特性を左右する重要なポイントとなります．

また，その材料をアクチュエータやセンサなどに採用した場合，構造と照らし合わせて，どの部分にどのような応力が発生するのか，強度は十分かなどを検討することになります．そのため，今度はシミュレーションによる解析が必要となります．いくつかの構造解析用シミュレーション・ソフトウェアがありますが，これを使って最適な設計値を決定していきます．このときたいせつなことは，シミュレーションはあくまでも目安にすぎないということです．実際の特性どおりになることは，まずありません．

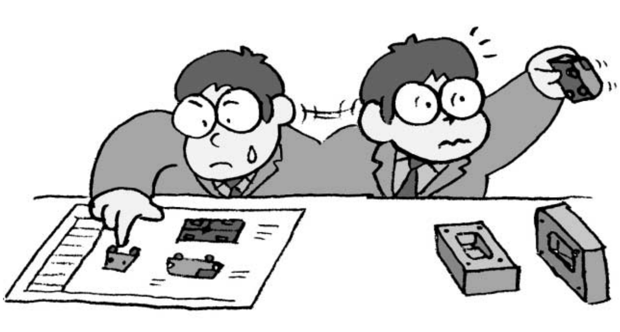


図11　製造できなければ意味がない

どんなに設計が理想的であっても，それが実現(製造)不可能なものでは意味がない．そのため，設計者は製造技術を理解していなければならない．

MEMSは，設計と試作の繰り返しによって製品が生まれます．そのため，設計者は製造技術を理解していなければなりません．設計が理想的であっても，それが実現(製造)できなければ意味がありません(**図11**)．また，量産性やコストを考えて設計しないと，製品にはなりません．

次にMEMS設計者に求められることは，その製品の目的と機能を具現化することです．センサであれば，それによって検出されるものとして，温度や圧力，加速度などがあります．例えば温度の場合，その温度をどのような方式でどのような信号に変換し，検出するのかを検討します．温度の検出方法はいろいろあります．赤外線を感知する方法，抵抗値の変位で検出する方法などです．また，その信号をアナログ値からディジタル値に変換するなど，その製品の目的に合った検出方法を考えなければなりません．

MEMS設計者は，従来からある技術はもちろんのこと，新しい技術にも目を向け，幅広いスキルを習得し，それを製品に生かすことがもっとも重要だと筆者は考えます．

● 図面どおりにシリコンを加工するのが難しい

MEMSの設計者にとってもっともプレッシャとなるのは，まったく新しい製品への取り組みだと思います．既存製品のマイナ・チェンジや後継品種の設計は(もちろん楽ではないのだが…)，基本的なコンセプトが固まっており，ある程度方向性は決まっています．しかし，まったく新しい方式のセンサなどを開発する場合，基礎となるデータや情報がほとんどありません．社内でプロジェクト体制が組まれ，製品化の日程に基づいて，各部署が開発を進めます．筆者個人がつねに厳しいと思う点はこの開発スケジュール，すなわち時間です．MEMS製品は，その製品の心臓部(コア)となる部品であることが多く，その開発日程は非常にタイトです．そして，この部分ができ上がらないと，製品

そのものが完成しません．

筆者はこれまで，シリコン・ウェハを用いた圧力センサやインクジェット・ヘッドの開発に従事してきました．入社2，3年目からは自分で設計したデバイスを自分で試作・評価し，さらに改良し，製品コンセプトに合致したセンサを作り上げてきました．このとき，図面どおりにシリコンを加工することにたいへん苦労しました．

シリコン加工にはいろいろな方法があります．応力に対するセンサの強度を上げるため，マイクロクラックのない加工方法を選定したり，接合層の厚さが一定になる方法を考案したりと，設計業務だけでなく，図面どおりに加工するプロセスを考えるのに多くの時間を費やしました．

しかし，自分の設計したデバイスが採用され，製品化されたときの喜びや達成感は絶大です．製品が完成するたびに感動します．自分の考えたデバイスが製品となり，特許としてその技術が残ります．今まさにMEMSを開発している設計技術者は，この喜びを糧に新製品の創出に力を注いでいるのだと思います．

● 2次元で考えるのではなく，3次元で構造をとらえる

MEMSの設計は，まず，机上における構造検討から始まります．この構造検討の多くは2次元的な図によって行われています．この2次元の構造検討には抜けが生じることがあります．MEMSの立体構造を平面的にとらえていると，大きなミスを引き起こします．最近では3次元CADツールが普及しており，これを活用することでトラブルを減らせます．机上での構造検討は，文字どおりさまざまな角度から行うことをお勧めします．

次に，設計者として行うべきことは，自分で設計したものを自分で作ってみることです．ひとりでできないときは，製造技術者の助けを借りながら作ります．実際に自分の手で作ることによって，設計した製品のどこに製造上の問題があり，どうすればそれを解消できるのかが見えてきます．設計したとおりのものを実現(製造)できない場合は，設計変更が必要になります．MEMS設計者は，机上で図面を引いて終わるのではなく，オールマイティな技術者になるように努力しなければならないと思います．

最後に，これはどの技術分野でも同じことだと思いますが，ネットワークを活用することが重要です．さまざまな分野の人と接し，そこから得た情報を活用することです．そのためには，学会や講演会，産学連携の場をおおいに利用するべきだと思います．優秀な技術者は，多くの講演の場に参加しています．そこで出会う同業種，異業種の技術者や大学の先生などと関係を密にすることで，みずからの知識の幅を広げられます．

MEMS応用製品は，今後，多くの分野で市場を拡大していくと思います．この技術は，ものをコンパクトにするだけでなく，安全や便利さなど，生活を豊かにしてくれる技術であると筆者は信じています．

そうま・しんいち
富士電機システムズ(株)

私の推薦書

「わかりやすい半導体デバイス 改訂2版」



三菱電機 技術研修所 編
オーム社 刊，ISBN4-274-03548-4，
B5判，218ページ，
2,940円，2001年3月

本書は半導体デバイスの入門書です．昨今，日本も欧米と同様にバルク・マイクロマシンだけでなく，サーフェイス・マイクロマシンも発展しつつあります．筆者はもともと化学を専門としていました．入社当時からバルク・マイクロマシンの製品を開発してきましたが，現在ではサーフェイス・マイクロマシンを応用した製品の開発にも参加しています．

筆者は電子デバイス技術がそれほど得意ではありませんでした．これまでにいろいろな書籍を読んできましたが，本書は，最近筆者が読んだ書籍の中では，もっともていねいに電子デバイスが解説されており，さまざまなデバイスの概要が簡潔に記述されているように思いました．また，機能や理論だけでなく，その製造方法や故障などのメカニズムについても解説されているので，設計者にも製造技術者にも役に立つ書籍だと思います．基礎的なところが多いため，この分野のエキスパートにとっては物足りないかもしれませんが，センサと回路を1チップ化させる技術が主流の今日では，ぜひ，一読していただきたい内容だと思います．

具体的な書籍の内容としては，半導体デバイスの特徴や基礎，バンドモデル，pn接合，トランジスタの製造方法，LSIについて紹介しています．さらに，半導体レーザやマイクロ波デバイスなどの紹介もわかりやすくまとめられています．本書を足がかりに，専門的な書籍へと進んでいけばよいのではないでしょうか．

筆者は書籍を選ぶとき，たいてい図解付きのものを選びます．理由は，あまり活字を読んでも理解できないからです．人によってさまざまかと思いますが，自分の理解しやすい書籍を選ぶのがいちばんだと思います．

12 システム開発者

西村芳一

システム開発者とは

システム開発者は，商品企画が固まると，与えられた仕様や競合他社の動向を見ながら商品化設計を行う．まず，どのような処理ブロック構成にするか，どのようなテクノロジを用いるかを決める．ブロックの構成に問題がなければ，どの部分をLSI（ASIC，FPGA）化するか，どの部分をソフトウェアで処理するかといった役割分担を決める．その後，各担当者に詳細設計を依頼し，それらをつなぎ合わせて全体の動作に問題がないかを調べ，そしてコスト的に問題がないかを確認し，競争力のある商品としてまとめる．

● 商品コスト，最新技術，開発期間を念頭に

システム開発者は，商品を実現するために，限られたハードウェア資源やソフトウェア資源を有効に使わなければなりません．また，安く商品化しないといけないので，さまざまなノウハウとセンスを必要とします．知識としては，ソフトウェア設計，ハードウェア設計，LSI設計など，商品に関係する広範囲の技術を知っている必要があります．

細部にわたる技術はかならずしも重要ではありません．与えられた仕様を見たときに，自分が持つ知識の中から，最適だと思われる実装方法を選び出し，開発を行います．この最適化の方向性には，以下の三つの項目があります．

● もっとも商品コストが下がるのはどの実装方法か

自分の得意分野がソフトウェアだからといって，すべてをソフトウェアで実現すると，ハードウェアから見たときとてもむだな設計になることがあります．ソフトウェア，ハードウェアの垣根を越えて商品化コストがもっとも下がる最適な組み合わせを探ります．そのためには，ハードウェア設計とソフトウェア設計の両方の経験が必要になります．

● どのような最新技術を導入するか

他社にない特徴を出し，優位性を確保するためには，まねのできない最新技術を導入するのがいちばんです．しかし，持っている知識をそのまま自分の商品に当てはめることは，一般にはできません．ここがシステム開発者の腕の見せどころです．導入リスクを考えながら選択・適用します．つねに世界の技術動向に目を向け，感度を上げておく必要があります．

● いかに開発期間を短縮するか

商品開発サイクルは短くなる一方です．社内のエンジニアの実力に合わない難しい技術を選択すると，いつまでたっても商品ができ上がりません．かといって，あまりにも従来技術に固執すると，競争に負けてしまいます．社内のエンジニアの実力を知ることが必要です．また，開発期間を短縮するためには，設計ツールの知識や経験も必要です．

最終的な商品開発が成功したかどうかの判断は，いかに商品が売れて市場に認められたかで決まります．いくらすばらしい技術を導入しても，売れなければ何の意味もありません．したがって，市場での競合他社の商品についての知識も欠かせません（**図12**）．



図12　競合他社の商品を研究する
いくらすばらしい技術を導入しても，売れなければ何の意味もない．したがって，市場での競合他社の商品についての知識も欠かせない．

システム設計では特許も重要です．関連する技術に対してどのような特許が存在し，どのように抵触するかの知識が必要です．新しい技術を導入する場合，特許の問題を避けて通ることはできません．また，自身も新たな特許を取得し，他社に対する優位性を確保しなければなりません．

そのほか，安全規格や不要放射ノイズの規格，環境規格なども知っておく必要があります．商品の海外展開に合わせて，こうした規格を考慮して開発することも重要です．

● 自分が担当している商品をどこまで好きになれるか

商品性を決めるのはシステム開発です．LSI設計やソフトウェア設計などで細部を個別に最適化することは可能ですが，システム全体の最適化の効果にはかないません．自

分の知恵や，ときには自分の人脈を駆使して競争力のある商品を作り上げるところに，ほかでは味わえない大きなやりがいを感じます．とくに，自分が開発した商品が話題になり，多くの顧客に購入してもらえたとき，「このしごとをやっていて良かった」と強く感じます．

そのためには日々，新しい技術を導入しなければならないのですが，それらを習得するためにはふだんからの勉強が欠かせません．なお，新しい技術の導入にはかならずリスクがつきまといます．なんらかのトラブルが発生するのを覚悟のうえで，採用しなければなりません．従来技術を使っていれば安心ですし，トラブルの発生も少なくてすみますが，それは進歩ではなく退化です．システム開発者には，トラブルを恐れずに飛び込むだけの勇気が必要です．

何よりも，自分の作っている商品が好きでないと続きません．システム開発者がもっともよりどころとしているのは，自分が担当している商品が好きだという気持ちです．それがあれば，新しい技術の勉強や知識習得，さまざまな苦労が苦痛ではなく楽しみに変わります．

● 深い分野にはまりこんで，全体を見失わない

システム開発者を目指すのであれば，早いうちからソフトウェア，ハードウェアなどの個別の深い分野にはまり込まず，全体を見失わないことが肝要です．もちろん新人のときは何もわからないので，当初は商品の一部の設計しか担当できません．しかし，その状況に満足せず，その商品にかかわるすべてのことに興味を持つことがたいせつです．

そして機会があれば，得意分野がソフトウェアであればハードウェア開発から，ハードウェアであればソフトウェア開発からというように，ほかの技術分野からのアプローチを経験するとよいでしょう．自分が開発を担当している部分が商品全体の中でどのような位置づけなのか，また，もし自分がシステム設計を行うのであればどこを改善できるのか，といったことをつねに考えるようにしましょう．

システム開発には広範な知識が必要です．自分の専門分野以外のところでわからない技術分野があった場合には，みずから勉強して習得しなければなりません．知らない技術をいかに地道に身に付けていくかが，後々，大きな差となって現れます．この資質が，システム開発者にとってもっとも重要なものだと思います．さらに，この勉強をつらいと感じるのではなく，楽しくなるようにできるかどうかが重要です．何でもそうですが，自分の好きなことであれば勉強は苦になりません．

それから，技術とは直接関係ありませんが，広い人脈を持つことはとても重要です，いくら勉強しても，個人の力には限りがあります．もしわからないことがあったとき，気軽に相談できる人を知っていたら，早期に問題を解決できたり，適切な判断を下せます．さらには，その人脈を生かして，とうてい自社だけでは実現できないような高度な技術を使った商品の開発も可能となります．機会を見つけて，積極的に人の輪を広げていきましょう．

にしむら・よしかず　（株）エーオーアール

2

私の推薦書

「新・管理者の判断力――ラショナル・マネジャー」



Charles H. Kepner，Benjamin B. Tregoe 著，上野一郎 監訳
産能大学出版部 刊，ISBN4-382-04851-6，19.5cm×14cm，361ページ，2,730円，1985年2月

筆者の本棚には多くの技術書がありますが，あらためて「システム設計者向け」といっても該当するものがなく，途方にくれてしまいました．筆者自身は電気の本だけでなく，趣味として物理関係の本も多く読んでいます．それらの中から本業のおもしろいヒントを得ることがよくあります．したがって，物理関係の本を選ぼうとも思ったのですが，それではあんまりなので，ここでは「新・管理者の判断力」を紹介します．

これは，筆者が会社で強制的に参加させられたKT法（本書の著者であるCharles H. KepnerとBenjamin B. Tregoeが考案した論理的思考法）の研修の最後にもらった本です．初めはあまり期待していなかったのですが，読んでみるとけっこう思い当たることが書いてあります．有能な管理者（ラショナル・マネージャ）とそうでない人はどこが違うのか，といったことが書かれています．このラショナル・マネージャとシステム開発者の間にいろいろと重なる部分があります．

日ごろの業務の中でトラブルにみまわれたとき，その対処のしかたによってシステム開発者（あるいはマネージャ）の実力の差がはっきりと現れます．トラブルを瞬時に解決できる人もいれば，解決に1ヵ月かかる人もいます．あるいは，せっぱ詰まって，その場限りの対症療法（カット・アンド・トライなど）に走る人もいます．

システム開発者が開発業務を進める中で，いかにスマートに対処するかのヒントが本書には書かれています．多くの例題を読み進めるうちに，そうしたヒントがつかめるかもしれません．本来であれば，KT法そのものについてもきちんと習得するべきなのかもしれませんが，単なる読みものとしても気軽に読めるので，お勧めです．

13 開発購買

沖祠照荏

開発購買とは

購買部門本来のしごとを大きく分類すると，契約業務と調達業務に分かれる．その中で取り扱われる材料は，原材料と製品に分かれ，さらに半導体・電気系と機構系に分類される．ここでは電子技術者を想定し，半導体・電気系のデバイスを対象とする．これから述べる「開発購買」の業務は，契約業務よりさらに上流の新製品開発時に活躍する．具体的には取引メーカやデバイスの選定で設計技術者をリードし，その商品の性能や機能，そしてもっとも重要なコストを構築する役割を果たす．

● もっともお金を使う部門なので法律の知識が不可欠

開発購買に限らず，購買部門の担当者全般に求められるのが法律の知識です．これは，製造業においてもっともお金を使う部門が，材料を購入する購買部門だからです．そのため，その契約や調達には厳しい目が向けられます．とくに商法や下請法(下請代金支払遅延等防止法)，公正取引法，最近では個人情報保護法といった法律を理解しておく必要があります．ここ数年，各企業ともコンプライアンス(法令遵守)を強化しているため，外部と金銭的な関係が深い購買部門はとくに注意が必要です．これだけでもたいへんですが，さらに最近では欧州からの厳しい要求であるRoHS(ローズ)指令(2006年7月1日に施行する電気電子機器への特定有害物質規制)などもあります．もし，法律違反が判明して製品の回収命令が出れば，企業が被る損害は巨額になり，その存続さえも危うくなる可能性があります．

開発購買の担当者には，デバイスに対する深い知識が求められます．デバイスの性能や機能(数値や特徴)といった項目はもちろんのこと，コストや品質，供給安定性についても熟知する必要があります．半導体デバイスについて言えば，原材料や加工工程，生産リード・タイム，生産場所(国)，さらに景気動向と業界動向などを把握できていないと，このしごとは務まりません．電子デバイスはさまざまな市場動向の変化によって材料や工場のラインの過不足が顕著に表れ，流通量に大きな影響を与えます．しっかりした専門知識があれば，事前に変化を予測し，対策を立てられるわけです．何の備えもなく，ただ漫然とデバイスを購入していたら，供給不足などであっという間に生産が止まってしまいます．

開発購買の職務に求められるスキルについて述べます．「開発購買」と名まえが付く以上，開発技術に関するスキルが必要です．本来の目的は，商品開発段階におけるデバイス選定によるVE(value engineering；右掲のコラムを参照)ですから，商品の企画意図や設計内容にも通じていなければなりません．設計技術者に信頼される技術力とサポート力が求められるしごとで，どちらかと言うと，深さよりも広さが要求されます．要は，専門家が見落としやすい問題をうまくフォローし，アドバイスしながらデバイス選定のプロセスを円滑に進める縁の下の力持ちです．

次に必要なスキルは語学です．最近ではグローバル化によって海外部品の調達が必須になっています．英語や中国語を中心に，語学力の強化が進められています．例えば，半導体については英語が中心になります．これまでは販売代理店である商社などがメーカとの間に入って通訳を行っていましたが，スピードが求められたり，より緊密なパートナシップを築くためには，購買担当者が直接メーカと対話する必要があります．それには日本語よりも英語のほうが効果的です．「お客さまでございます」という態度が通用した時代は，すでに終わっています．

また，プレゼンテーションのスキルも重要です．先に述べたように，コスト交渉や供給交渉ではメーカとのパートナシップの構築が重要になります．そのため，商品開発に



図13　開発購買担当者にはプレゼンテーション能力が必要
相手に与える印象が大事．スピーチ力やユーモアも要求される．

関する情報を開示しなければなりません．要はどれだけ自分の会社の価値を相手に高く売り込めるかが問題となります．会社の代表ですから，それなりに重圧のかかるしごとです（**図13**）．

● 相手は百戦錬磨のマネージャや経営者

このしごとでいちばん苦労するのは，取引先との交渉です．買い手側がヒラの担当者でも，売り手側は社長や専務，部長といった百戦錬磨の人々が出てきます．話題もほとんど業界や会社の近況，経済見通しで，本題はほんの少しです．そこでは広い知識が要求されますし，服装や態度，ことばづかいにも気を遣います．交渉も，押したり引いたりのころ合いと落としどころが難しいです．これといった教科書はないので，実地でコツをつかむしかありません．

次に苦労するのが人脈作りです．どれくらい他社の幹部を知っていて，困ったときに無理を聞いてもらえるかがポイントとなります．とくに半導体はリード・タイムが16週と長く，海外企業も多いので，外国人トップに対して直接要望を出せるかどうかで結果がまったく違ってきます．

では，「このしごとのやりがいは何か」ですが，もちろん海外も含めて広い人脈ができることや，業務でさまざまな場所に出張して視野や知識が広くなるといったこともあります．しかし，ほんとうのやりがいは，商品企画の意図をくんで，機能や性能，コストを技術部門とともに作り込み，ヒット商品を生み出したときだと思います．協力してもらった取引先に注文量でお返しができたとき，営業担当者の喜ぶ声や表情に，しごとへの満足とやりがいを感じます．

● 企業経営の一端を担う戦略部門へ

開発購買のしごとを目ざす人には，バランス感覚が要求されると思います．交渉力はもちろんのこと，企画力や技術力が必要なうえ，金銭感覚，すなわちある種の経営センスも求められるからです．購買部門は，かつての単なる資材購入部門から，現在は企業の経営の一端を担う戦略部門に変化しています．まさに新しい人材が求められている部署なのです．現在，設計技術者で，営業経験やサービス経験があり，日ごろから世の中の変化が気になっているような人は，十分に開発購買になれる素質があると思います．

開発購買のしごとは，技術部門や資材部門，品質管理部門，そして工場を引っ張るリーダシップを必要とします．また，取引先との交渉ではタフさや頭の回転も要求されます．とにかく何か人を引き付ける魅力が持てるように，たえず自分を磨いていかなければなりません．簡単ではありませんが，やりがいは十分にある業務です．会社にとっても，自分にとっても….

おきほこら・しょうじん
開発購買担当

私の推薦書

「新・VEの基本―価値分析の考え方と実践プロセス」



土屋 裕 監修，産能大学VE研究グループ 著
産能大学出版部 刊，ISBN4-382-05451-6，
A5判，234ページ，2,310円，1998年5月

この書籍は，全般に渡ってVE（value engineering）について書かれています．簡単に言えば，商品の企画（機能，性能）を変更せずに価値を高めるにはどうすればよいか，どういう手法が用いられるか，ということが書かれています．基本は「商品価値（value）＝機能/コスト」です．商品価値を上げるならば，機能を増やすか，コストを下げるか，あるいはその両方を実現して大きな価値を生み出すか，となります．この本では推奨していませんが，最近では機能を少し絞り込んで大きなコスト削減を実現し，価値を上げることもあるようです．

現在の市場はこの公式や手法だけでは対応できない複雑な要求や予想外のヒットがあったりと，たいへんな時代になっています．しかし，このような時代こそ基本に立ち戻り，誤りや思い込みを修正する必要があると思います．また，VEにむとんちゃくな設計技術者を誘導する事例集としても本書は参考になります．

この本は，生え抜きの購買担当者よりも，むしろエンジニアから開発購買へ転身する方に読んでほしい1冊です．VEということばは聞き慣れていて，取り組む内容もある程度理解しているのでだいじょうぶと思っている人が多いようですが，実際にVEを実践するのは開発に携わる設計技術者です．その説得は一筋縄ではいきません．

他人には，しごとの価値をできるだけ定量的な表現で伝える必要があります．そんなとき，この本の手法や表現方法を用いれば，人を説得しやすくなります．どのような形で相手に価値を伝えるかの基本がわかる1冊です．開発購買の業務評価では，コストダウンがもっとも大きなウェイトを占めます．経験値だけでは判断を誤りやすい部分を補い，バランス感覚を養うためにも，本書の熟読をお勧めします．